



Continuous Diagnostics and Mitigation (CDM) Overview Training

March 17 2014



Homeland
Security

Continuous Diagnostics and Mitigation (CDM)

Module 1 – CDM Overview
Part 1 – Program Objectives



**Homeland
Security**

Module 1: CDM Overview

- Where in the training sequence does this module fit?



**Homeland
Security**

Learning Objectives

- At the conclusion of this module, the participants will be able to:
 - Describe the purpose and benefits of CDM
 - Describe the capabilities provided through CDM and how each are related
 - Describe the importance of each capability to information security
 - Describe how CDM represents a paradigm shift from the current approach of manual control testing and reporting
 - Describe the roles and responsibilities of various partners in the CDM program



**Homeland
Security**

Why CDM?

- A recent report from CSIS¹ found that CDM stops 85% of cyber attacks by:
 - Searching for, finding, fixing, and reporting the worst cyber problems first in near-real time
- It will also enable System Administrators to:
 - Respond to exploits at network speed
 - Fulfill A-130 responsibilities as intended
 - Implement NIST Publications on Continuous Monitoring (800-137 and parts of 800-37)
 - Use strategic sourcing to lower costs

¹James A. Lewis, [Raising the Bar for Cybersecurity](#). Washington, DC: CSIS, 2013.



Why CDM? (Cont.)

- According to the CSIS report:
 - 75% of the attacks use known vulnerabilities that could be patched;
 - More than 90% of successful attacks require only the most basic techniques; and,
 - **96% of successful breaches can be avoided** if the victim puts in place simple or intermediate controls.



Homeland
Security

What is the Problem?

- **Every Three Days (on Federal networks):**
 - Trillions of cyber events
 - Billions of potentially defective hardware, software, and account changes
 - Millions of attempted attacks at Internet speed
 - Thousands of new flaws introduced
 - Hundreds of successful attacks
- **Every Three Months:**
 - Over 10,000 successful attacks
 - An unknown number of these attacks are repaired
 - Terabytes of data are stolen
 - Over 7,200 reports are written²
 - Hundreds of labor hours are wasted
- **Every Three Years:**
 - Thousands of assessments and other reports are written and issued. Each:
 - Requires 3 to 9 months to prepare;
 - Is out of date the moment it is printed; and,
 - Provides only a snapshot in time vs. real-time identification and mitigation of problems.

²Office of Management and Budget, Memorandum 02-01: Guidance for Preparing and Submitting Security Plans of Action and Milestones. Washington, DC: OMB, 2001.



Current Fix...

- **Short answer: Plans, Reports, and Manual Audits**
 - 3-month to 3-year remediation plans³
 - Triennial reports currently required by regulation
 - Manual audit and oversight

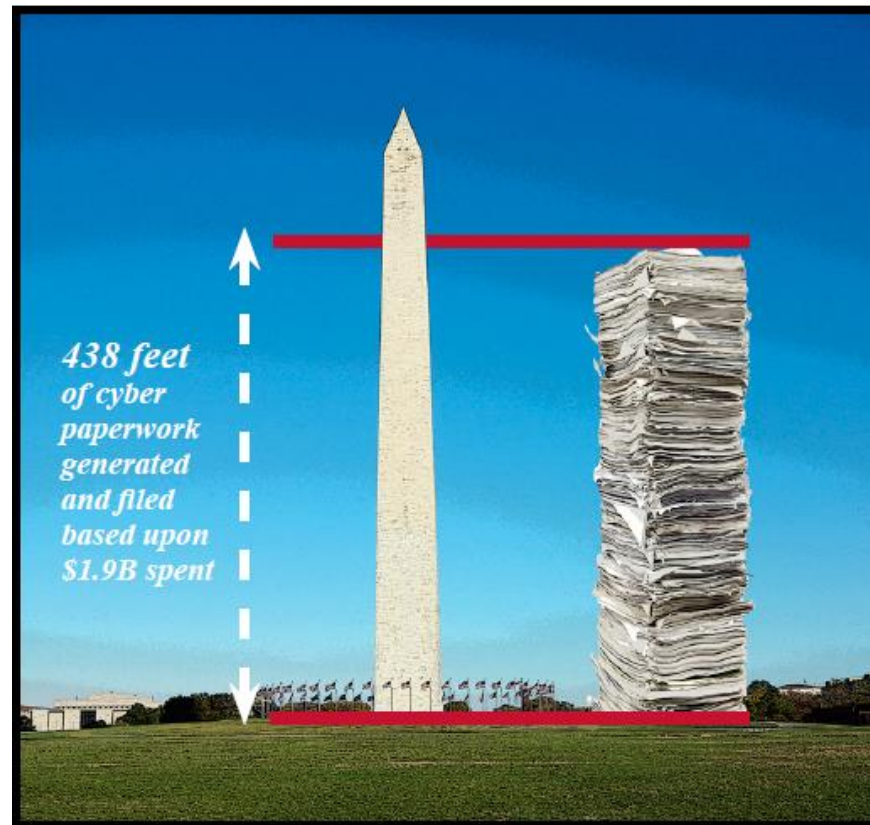
³ Office of Management and Budget, [Circular A-130, Appendix III: Security of Federal Automated Information Resources](#), Washington, DC: OMB, 2000



**Homeland
Security**

...the Results

- We estimate that these manual plans, reports, and audits cost between \$600M and \$1.9B a year, at a cost of \$1,400 per page



*438 feet
of cyber
paperwork
generated
and filed
based upon
\$1.9B spent*



**Homeland
Security**

In Other Words...



- **Encompassing:**
 - Manual audits
 - Manual Plan of Action reports
 - Manual Cyber Scope reporting
 - Manual Annual Testing
 - FISMA portion of financial statements



**Homeland
Security**

END OF SECTION

Continuous Diagnostics and Mitigation (CDM)

Module 1 – CDM Overview

Part 2 – CDM Scope (Capabilities)

Defining CDM Scope and Methods

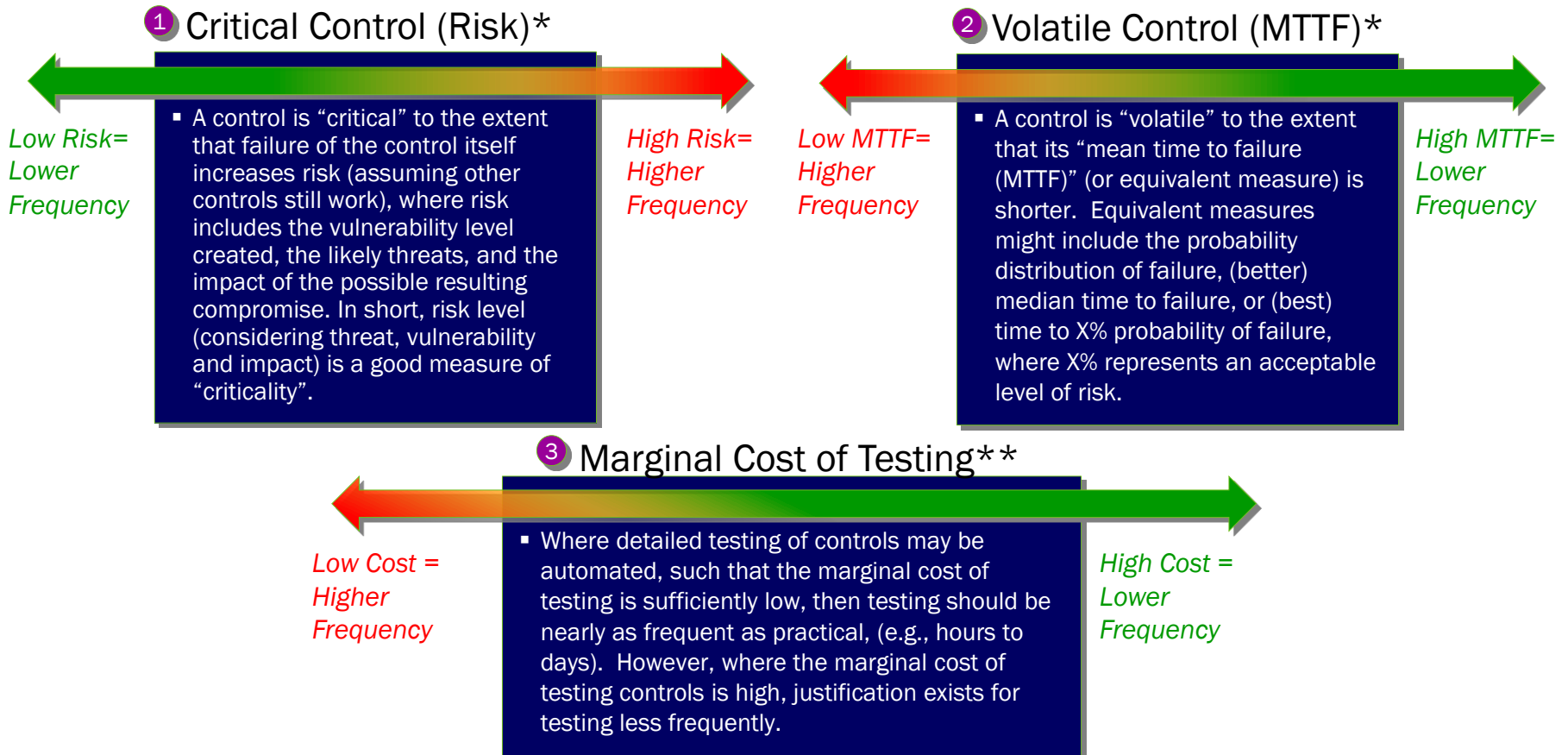
- As Information Security and Continuous Monitoring (ISCM) programs were piloted and successful, it became clear that a mature program needed to know:
 - How much should/could be included in an ISCM program?
 - In other words: How will we know when the ISCM program is appropriately and effectively implemented?
- **These questions entail a number of issues:**
 - How often should we test things to get the best return on investment? Especially things that cost a lot to test?
 - Is there anything we shouldn't test?
 - What are the most important things to test?
 - Can we automate tests of operational and managerial controls, or just technical controls?

How Will We Know When an ISCM Program is Fully Implemented?

- Since CDM is intended to operate on networks with Federal data, the obvious answer is that the CDM program is implemented (or complete) **when it can automatically test as much of the NIST SP 800-53 control set as possible and efficient.**
- As a result, DHS worked to define how to test as many 800-53 controls as possible. The initial assumption was that many would need manual testing, so DHS needed a good way to decide **how frequently to do expensive manual testing.**

Factors in deciding how often to test in order to get the best return on investment

DHS conducted an economic analysis to determine optimal test frequency. Three factors drive the outcome:



How Often Should We Test to Get the Best Return on Investment?

The economic analysis showed the following:

- Unless the marginal cost of testing is very low relative to the risk (estimated cost of failure), the item may not be worth testing.
- Where the marginal cost of testing is near zero, one should test as often as possible.
- If unsure how often to test, it is less risky to test a little less often, than too often as total cost goes up faster with marginally more testing (compared to too little testing).
- **Testing fewer high value results directly is more cost effective than testing many low value items that aggregate to produce a high value result.**
- This conclusion was acknowledged by NIST in 800-53 Rev4, which added the concept of a security capability.

What is a NIST “Security Capability”?

A (NIST) security capability means:

A collection (set) of security controls that work together to achieve an overall security purpose. (NIST 800-53 Rev4, p. 21.)

NIST notes that focusing on “security capabilities” improves/supports risk management:

- The failure of multiple controls, may not affect the overall security capability needed by an organization.
- Moreover, employing ... security capabilities allows an organization to ... determine if the failure of a particular security control ... affects the overall capability needed for mission/business protection.
- Ultimately, authorization decisions (i.e., risk acceptance decisions) are made based on the degree to which the desired security capabilities have been effectively achieved and are meeting the security requirements defined by an organization.

(NIST 800-53 Rev4, p. 21. Emphasis added.)

What are the Most Important (Highest Value) Items to Test?

- DHS needed to find a set of security capabilities that covered 800-53 controls.
- Initially DHS tried using NIST 800-53 control families, as they are currently widely used. However, the control families do not fit the NIST definition of a security capability:
 - There is no explicit purpose (or outcome) of each family.
 - Often controls in a single family have different purposes.
 - Often, to achieve a specific purpose, you need controls from several families.
 - The same is largely true of the CSC-20 “controls.”
- DHS analyzed all 800-53 controls, and grouped them by their purpose (the kind of attacks that they would address)
- This led to a definition of a CDM capability, which is consistent with the NIST definition, but has additional requirements.

What is a CDM “Security Capability”?

- A (CDM) security capability means a NIST security capability that has the following additional features:
 - The purpose is thwarting specific attack scenario(s)
 - Identification of the objects under attack in those scenarios (targets)
 - A Concept of Operations (CONOPS) for using information security continuous monitoring (ISCM) to detect and prioritize weaknesses of those targets for mitigation
- CDM Capabilities (as a group) are also intended to cover all important attack scenarios (and related) controls with minimal duplication. So for each capability we need clarification of gray areas and “edge cases” relative to other capabilities (differentiation).

Identifies all CDM Capabilities
Related Capabilities are grouped
into “Families”



Can We Automate Tests of Operational and Managerial Controls, or Just Technical Controls?

- It is often possible to automate tests of operational and managerial controls
 - Many operational and managerial controls are subject to direct automated testing
 - For example, if acceptable configuration settings are maintained in a database, and those desired state settings are compared to the actual settings, than the desired state specification:
 - Is an automated expression of policy (a managerial control).
 - Can be used to test whether policy is followed.
 - The act of using the automated policy to test compliance verifies that policy does exist, thus testing a managerial control.
 - Most operational and managerial controls are subject to indirect automated testing.
 - For example, NIST says: “risk acceptance decisions are made based on the degree to which the desired security capabilities have been effectively achieved” and that “the failure of multiple controls, may not affect the overall security capability needed by an organization.”
 - Thus, if the overall capability is being achieved, that may be considered an acceptable outcome, even if some managerial and/or technical controls fail. This is indirect testing. When the capability fails, root cause analysis is then needed to find the cause.

Phase 1 CDM Capabilities

- **Purpose** – Identify unauthorized and unmanaged devices that are likely to be used by attackers as a platform from which to extend compromise of the network.
- **Controls** – Collects controls related to configuration and supply chain management of hardware, including changes introduced during travel.
- **Targets** -- Attackable Hardware Devices including all IP-addressable devices (or equivalent) on a network, plus selected attackable sub-components (like removable media)
- **ConOps** – Maintain a list of authorized hardware and who manages it. Treat other hardware actually on network as a defect. Remove, authorize/assign or accept risk.
- **Differentiation** -- HWAM does not address **how well** software on the device is managed, but only that management is assigned. How well the software is managed is covered by software asset management, configuration setting (CCE) management, and vulnerability (CVE) management.



Sample Mapping to 800-53 Controls

Defect Types

ID	Defect Type	Determination Statement	Mitigation Options	Selected
F1	Unauthorized Devices	In Actual State but not in Desired State [See supplemental criteria in L2]	<ul style="list-style-type: none"> Remove Device Authorize Device OR Accept Risk 	Yes
F2	Unmanaged Devices	In Actual State and in Desired State but no “appropriate” manager assigned	<ul style="list-style-type: none"> Remove Device Assign Device OR Accept Risk 	Yes
F3	Non-Reporting Devices	In Desired State but not in Actual State	<ul style="list-style-type: none"> Restore Device Reporting Declare Device Missing OR Accept Risk 	Yes

Mapping to 800-53

Defect Type	Low Baseline	Moderate Baseline	High Baseline
F1	CM-02	CM-02 (1c), CM-03b, CM-03 (2), CM-08 (1), CM-08 (3a)	CM-02 (2), CM-03 (1a), CM-03 (1b), CM-03 (1d), CM-03 (1e), CM-08 (2)
F2	CM-08(4)		CM-08(4)
F3	CM-08a-1, CM-08b		

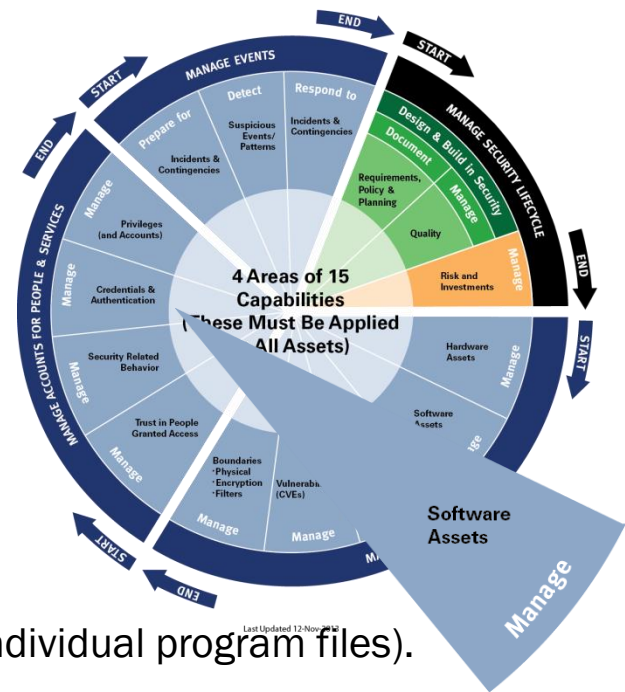
CM-02 BASELINE CONFIGURATION

Control: The organization develops, documents, and maintains under configuration control, a current baseline configuration of the information system.

Determination Statement: If the baseline (Desired State) is not maintained, it will be out-of-sync with the actual state, and defects will be reported.

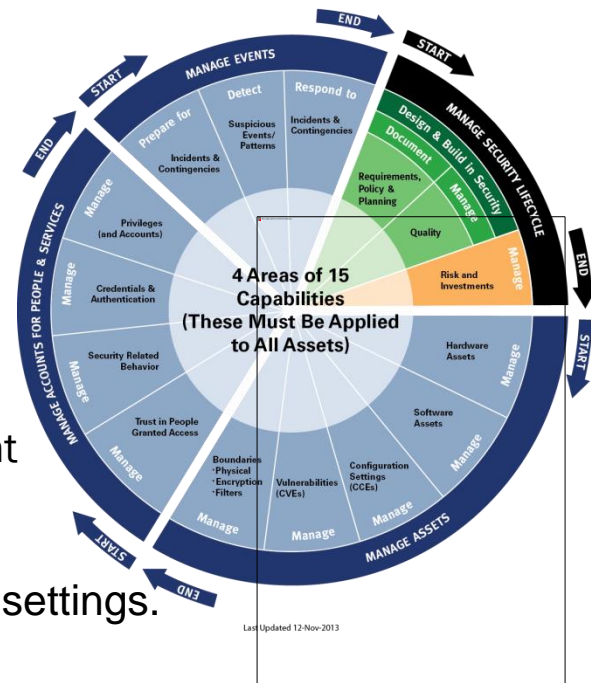
Software Asset Management

- **Purpose** – Identify unauthorized software on devices that is likely to be used by attackers as a platform from which to extend compromise of the network.
- **Controls** – Collects controls related to configuration and supply chain management of software, including changes introduced during travel.
- **Targets** – Software products (e.g., MS-Word) and executables (individual program files). Identify executables by their digital fingerprint.
- **ConOps** – Maintain a list of authorized software at both the product and executable level. Treat other software actually on network as a defect. Remove, authorize/assign or accept risk. Blocking unauthorized software can prevent many phishing attacks and zero day exploits.
- **Differentiation** – SWAM does not address how well software on the device is managed, but only that the software is authorized. How well the software is managed is covered by configuration setting (CCE) management and vulnerability (CVE) management.



Configuration Setting Mgmt

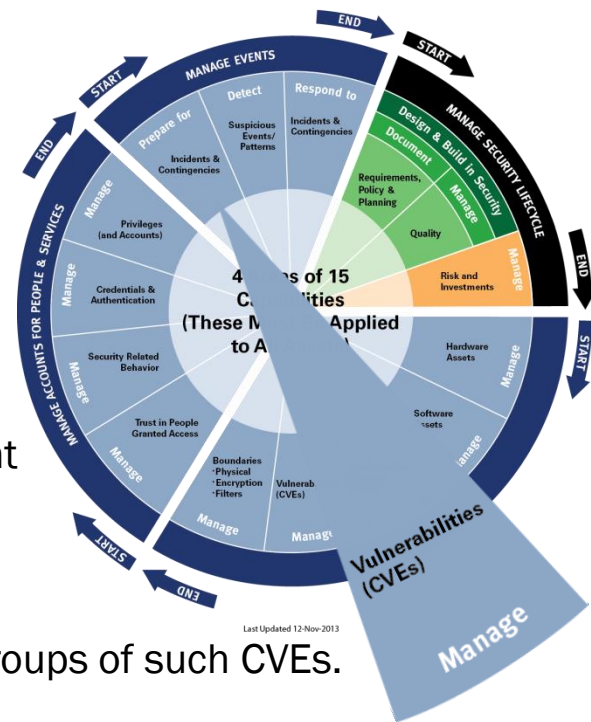
- **Purpose** – Identify configuration settings (CCEs) on devices that are likely to be used by attackers to compromise a device and use it as a platform from which to extend compromise to the network.
- **Controls** – Collects controls related to configuration management of software settings, including changes introduced by attackers.
- **Targets** – Individual Configuration settings, or groups of such settings.
- **ConOps** – Maintain a list of authorized settings. Treat weaker settings actually on network as a defect. Remove, authorize/assign or accept risk.
- **Differentiation** – Settings are often used as a means to support other capabilities, such a blocking certain software and/or granting/denying privilege(s).



At least initially, settings will be dealt with as one group, because most organizations manage CCEs through one process and the other capabilities have not yet been deployed. The CDM program is planning on reconsidering this over time, and moving checks that are currently performed with benchmarks and configuration compliance tools to more appropriate capabilities when deployed.

Vulnerability Management

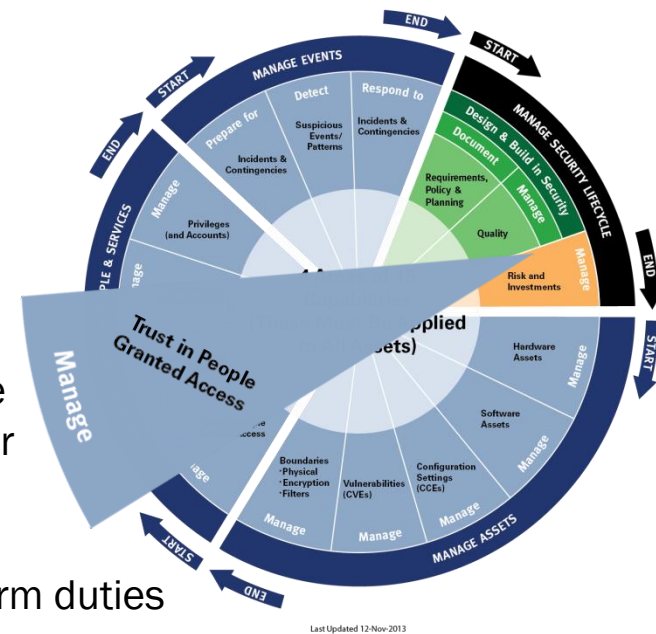
- **Purpose** – Identify vulnerabilities (CVEs) on devices that are likely to be used by attackers to compromise a device and use it as a platform from which to extend compromise to the network.
- **Controls** – Collects controls related to configuration management of CVE-like defects, most typically patches, including changes introduced by attackers.
- **Targets** – Individual CVEs and means to protect from them, or groups of such CVEs.
- **ConOps** – The National Vulnerability Database (NVD) provides a library of vulnerabilities mapped to vulnerable software. Upgrade the software to safer patches or versions, or accept the risk. CWE scanners identify poor coding practices (CWEs) that are directly associated with conditions that often manifest as vulnerabilities that are discovered and assigned a CVE.
- **Differentiation** – The term vulnerability is sometimes used loosely to mean any security weakness. In this case its use is limited to CVEs (and CWEs that cause CVEs).



Phase 2 Capabilities

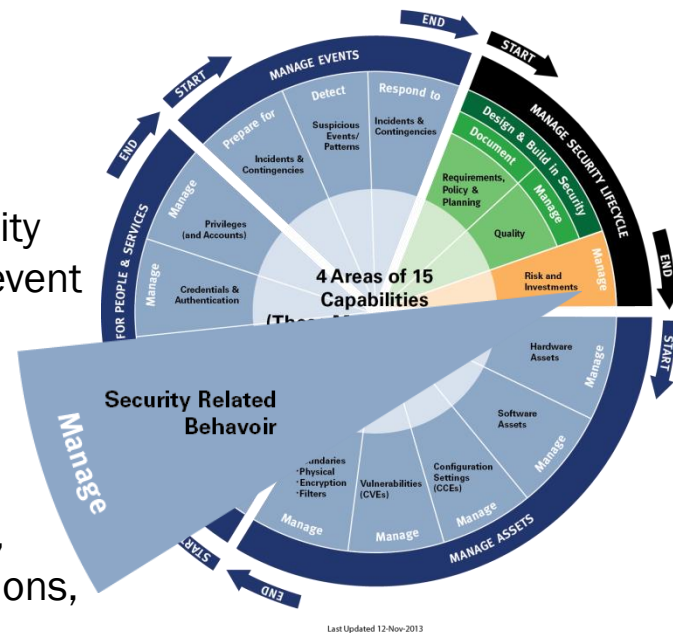
Trust Management

- **Purpose** – Ensure that untrustworthy persons are prevented from being trusted with network access to prevent insider attacks.
- **Controls** – Verifies that appropriate screening has been done recently enough to be reliable. Might be expanded to look for unreliable work behavior once trusted.
- **Targets** – People who are allowed (or to be allowed) to perform duties that require trust, including access to the network.
- **ConOps** – Track completion of screening processes (such as clearances, background checks, suitability reviews, etc.) designed to identify evidence of untrustworthiness.
- **Differentiation** – TRUST does not include proactive efforts to explain expected behavior and/or train persons how to behave reliably. Nor does it cover compromises due to careless (non-malicious) behavior. These are covered by the Behavior Management Capability.



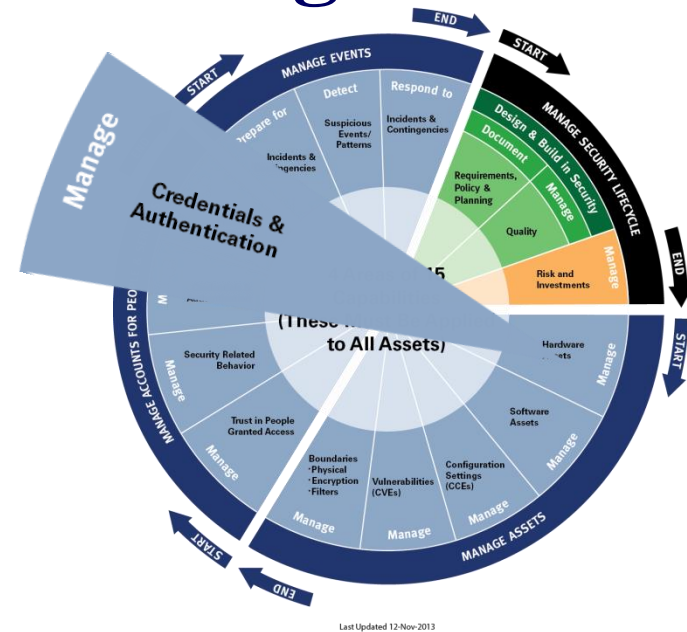
Behavior Management

- **Purpose** – Ensure that people are aware of expected security related behavior and are able to perform their duties to prevent advertent and inadvertent behavior that compromises information.
- **Controls** – Verifies that there is evidence of the ability to perform as expected based on reasonably current Training, Behavior/Use Agreements, Courseware and Skill Certifications, evidence of actual work behavior, etc.
- **Targets** – People who are allowed (or to be allowed) to perform duties that require secure behavior, including access to the network.
- **ConOps** – Track evidence (such as Training, Behavior/Use Agreements, Courseware and Skill Certifications, etc.) designed to specify and enable secure behavior.
- **Differentiation** – There is some discussion that certain BEHAVE controls belong in TRUST. DHS is working to clarify this differentiation.



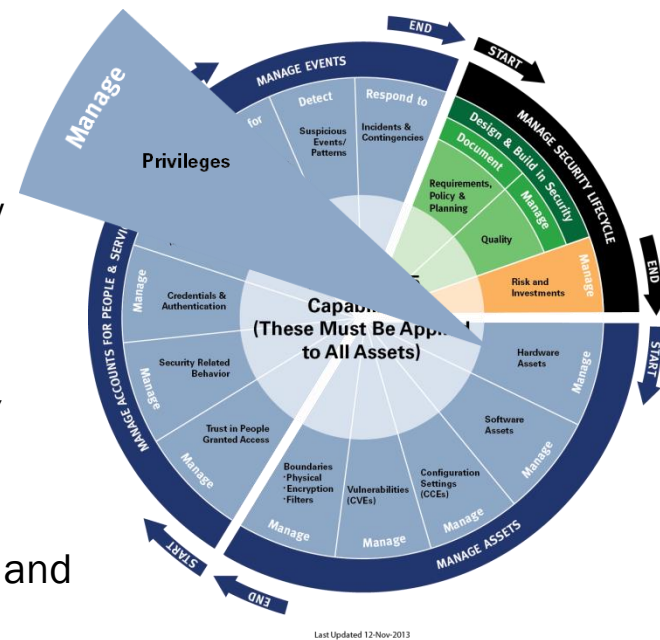
Credentials and Authentication Mgmt

- **Purpose** – Ensure that people have the credentials and authentication methods necessary (and only those necessary) to perform their duties to appropriately control access.
- **Controls** – Controls related to identification of the user, authentication, and non-repudiation. Assigning accounts to people and credentials, as well as device credentials.
- **Targets** -- Credentials and Authentication Methods for each user/device. Since credentials are for accounts, accounts are included here.
- **ConOps** – Computes the needed credentials and authentication methods from assigned user roles, and verifies that no extra credentials/methods are provided.
- **Differentiation** -- There is some thinking that certain accounts belong in PRIV. DHS is working to clarify this differentiation.



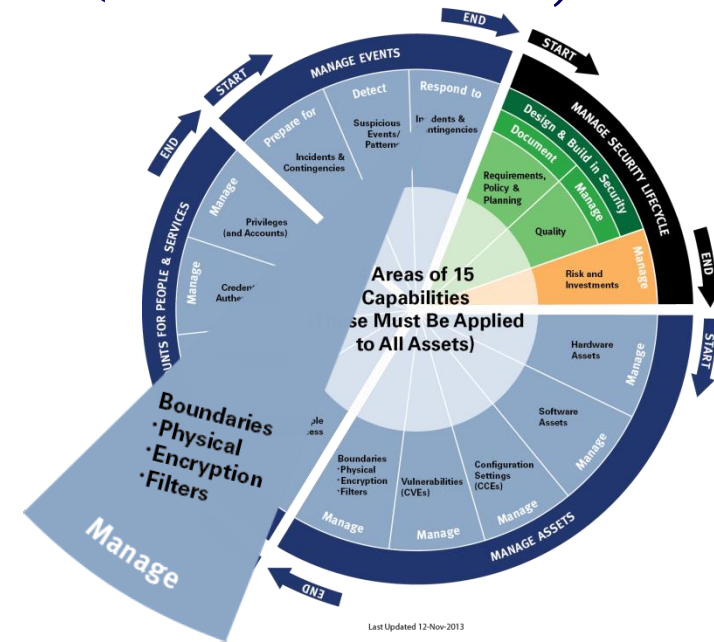
Privilege Management

- **Purpose** – Ensure that people have the privileges necessary (and only those necessary) to perform their duties, to appropriately control access.
- **Controls** – Controls related to permission to access objects/targets.
- **Targets** – Objects to which access is granted and the Users and Accounts which are granted that access.
- **ConOps** – Computes the needed privileges from assigned user roles, and verifies that no extra privileges are provided.
- **Differentiation** – There is some discussion that certain accounts belong in PRIV. DHS is working to clarify this differentiation. There are CSM settings that provide or deny privilege. They are on the boundary between CSM and PRIV.



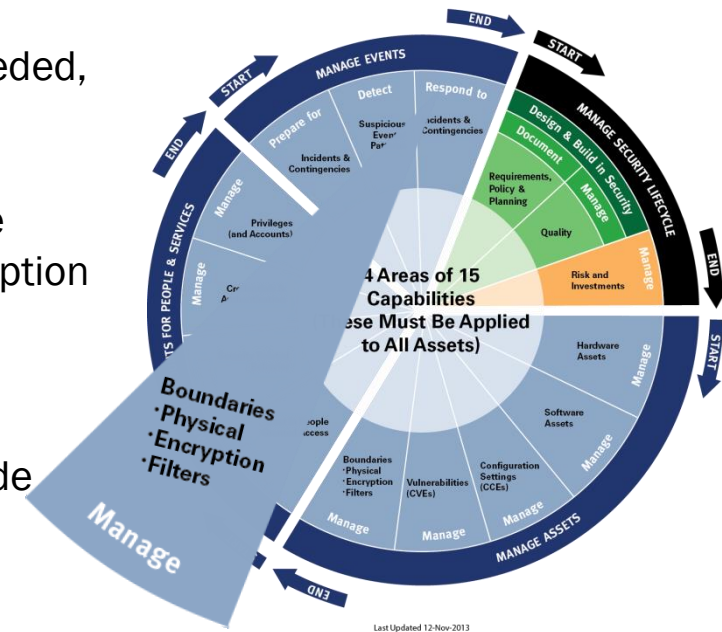
Network Boundary Control (N-BOUND)

- **Purpose** – Ensure that traffic in-to and out-of the network (and thus out of the physical facility protection) does not compromise security.
- **Controls** – Firewalls and content filtering, as well as their interaction with other controls.
- **Targets** – Network traffic
- **ConOps** – Ensure that firewalls, filters, etc. are in the desired state. Use attack graph modeling to identify potential weaknesses in the resulting network boundary defenses (as they interact with other capabilities).
- **Differentiation** – CSM can support firewall and filter configuration. The intent is that the overall system of controls does not leave unexpected attack paths to create unknown risk.



Virtual Boundary Control (V-Bound)

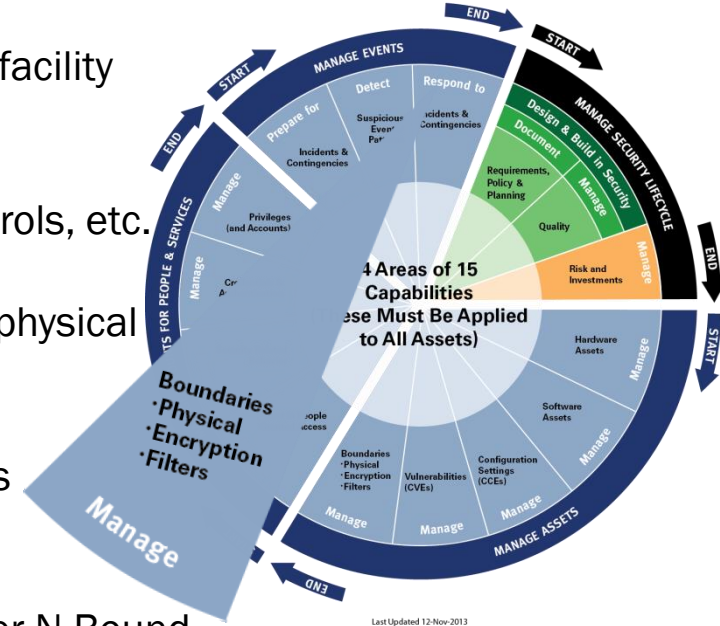
- **Purpose** – Ensure that information is encrypted when needed, whether in motion, or at rest.
- **Controls** – Encryption controls, primarily for virtual private networks and data at rest. In a cloud environment, encryption of all user data is often used to prevent one user from compromising another's data.
- **Targets** – Information on data links and computers outside of physical control. Information in environments shared with other users who are not trusted.
- **ConOps** – Ensure that encryption controls are in the desired state. Use attack graph modeling to identify potential weaknesses in the resulting network boundary defenses (as they interact with other capabilities).
- **Differentiation** – This control typically supplements/compensates for weaknesses in N-BOUND and P-BOUND.



Phase 3 Capabilities

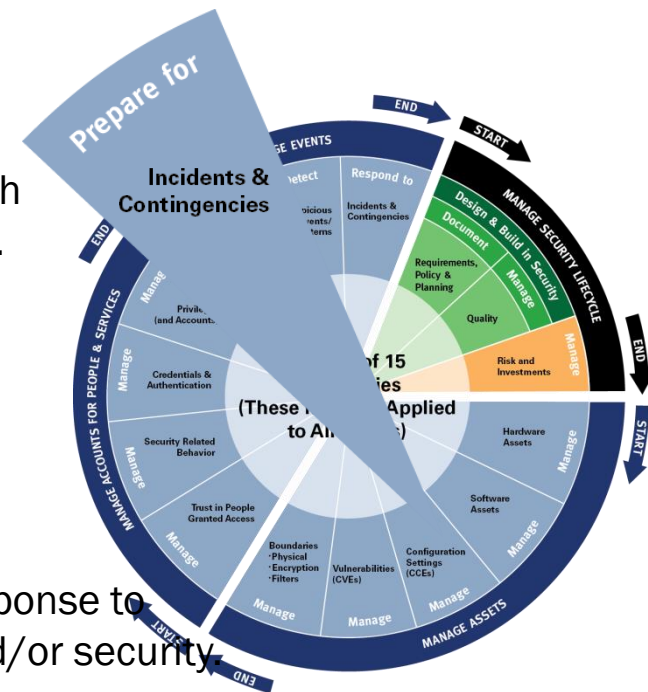
Physical Boundary Control (P-Bound)

- **Purpose** – Ensure that traffic into and out-of the physical facility does not compromise security.
- **Controls** – Physical access controls, logs, emanation controls, etc.
- **Targets** – People, Media, etc. coming into and out of the physical system boundary.
- **ConOps** – To extent possible, track physical access events and control, to ensure they are operating correctly.
- **Differentiation** – Emanation controls might better fit under N-Bound.



Prepare for Events (PREP)

- **Purpose** – Ensure that resources are in place to deal with both routine and unexpected events that can compromise security. These include cybersecurity incidents and contingencies (acts-of-god) like floods, earthquakes, etc.
- **Controls** – Contingency planning and incident response preparation controls.
- **Targets** – Events that compromise security that require a response to protect information and/or to restore system functionality and/or security.
- **ConOps** – Identifies the desired preparations (e.g., extra capacity, backups, etc.) and verifies that they are present (and ideally performing). Particularly relevant for preparations not used routinely.
- **Differentiation** – Event response covers the use of the desired preparations in an actual event. For events that occur routinely (using the prepared response), actual usage would likely obviate the need to test under PREP.



Last Updated 12 Nov 2013

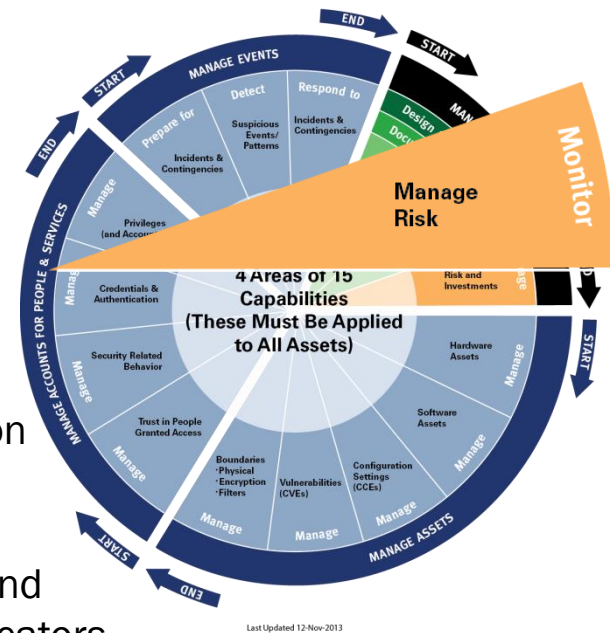
Respond to Events (RESPOND)

- **Purpose** – Ensure that both routine and unexpected events that can compromise security and require a response to maintain functionality and security are responded to as planned. Events include actual cybersecurity incidents and contingencies (acts-of-god) like floods, earthquakes, etc.
- **Controls** – Incident and Contingency Response Controls.
- **Targets** – Events that compromise security that require a response to protect information and/or to restore system functionality and/or security, as well as the Objects that are affected by such incidents.
- **ConOps** – Implements desired preparations (e.g., extra capacity, backups, etc.) and verifies that they perform well. Provides lessons learned to improve PREP.
- **Differentiation** – Events are identified in AUDIT.



Generic Auditing Monitoring (AUDIT)

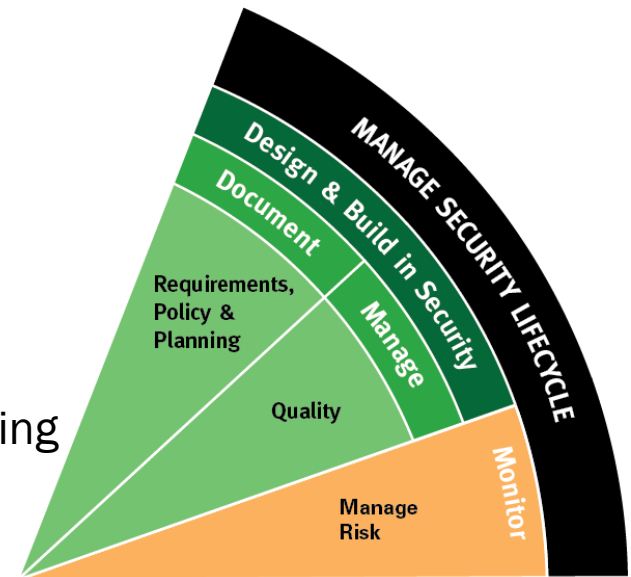
- **Purpose** – Identify routine and unexpected events that can compromise security. These include actual cybersecurity incidents and contingencies (acts-of-god) like floods, earthquakes, etc.
- **Controls** – Audit and related oversight controls.
- **Targets** – Events that occur on the network, and their impact on confidentiality, integrity and availability.
- **ConOps** – Uses various methods to correlate detailed events and track patterns of events to identify unexpected patterns or indicators of harmful activity.
- **Differentiation** – This capability provides feedback to most other capabilities, such as whether devices, software, users, credential, account, privilege, and data related behavior is as expected and related to necessary duties. Thus it is an advanced capability that supports those other capabilities.



Meta Capabilities

Meta Capabilities

- The remaining capabilities relate to implementing the overall security program – they are meta capabilities.
- In most respects these are done as part of implementing the other capabilities.
- For example:
 - Requirements for the CDM program are being defined by DHS/FNR
 - Plans for implementing CDM are based on a CDM architecture and ConOPS to be customized for D/As by the CMaaS contractors
 - Policy is largely established by the desired state specifications.
 - Quality management of CMaaS is done by DHS IV&V
 - Operational Security (Risk Management) is initially done through risk scores, and later supplemented by level 3 maturity metrics.
- If the capabilities are working, one can assume that these meta capabilities are working.



END OF SECTION

Continuous Diagnostics and Mitigation (CDM)

Module 1 – CDM Overview

Part 3 – Implementation

Topics

1. Defining Capabilities for Modular Implementation
2. How CDM Works
3. General CDM Paradigm Shifts
 - Desired State
 - Automation
 - Prioritization
 - New Roles (give examples)
4. D/A Technical Options (Decisions)
5. D/A Managerial Options (Decisions)
6. Impact on Network Performance
7. Security
8. New Technologies

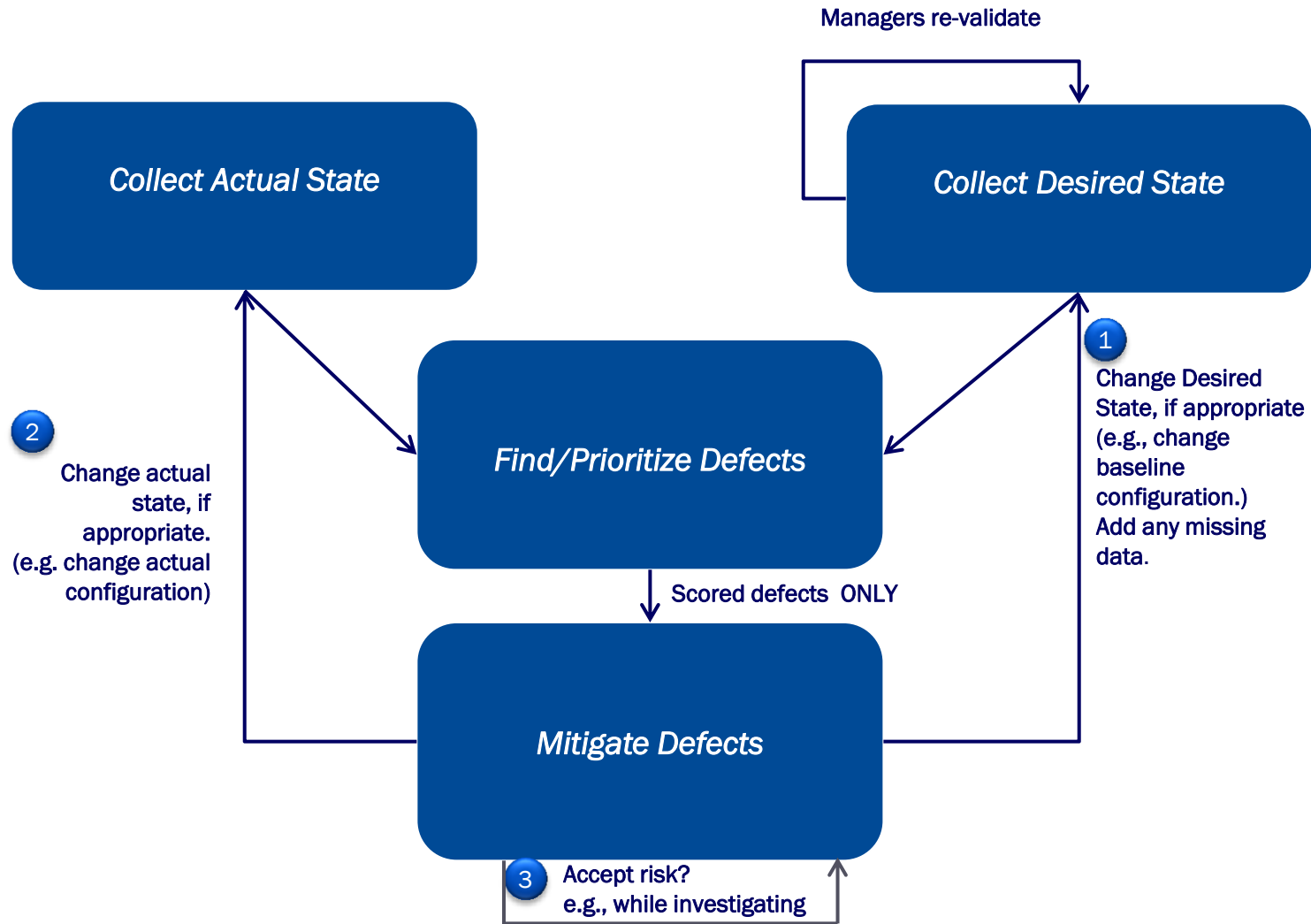
Modular Capabilities

- Each CDM capability should (to the extent possible):
 - Address a distinct attack type (have a distinct purpose)
 - Collectively, the CDM capabilities should protect from all relevant attack types.
- CDM capabilities interact and support each other. For example:
 - Knowing what devices you have allows you to know where to look for software.
 - Knowing what software you have allows you to know what settings you need to check.
- Being able to implement capabilities individually, or a few at a time, simplifies implementation.
 - The capabilities are designed to allow be implemented incrementally.
 - The performance metrics are designed to show incremental progress within each capability.
- So, you can eat the elephant one bite at a time.

How Will CDM Work?



CDM Generic CONOPS



Paradigm Shift

Definition: Paradigm

A set of assumptions, concepts, values, and practices that constitutes a way of viewing reality for the community that shares them, especially in an intellectual discipline.

(<http://www.ahdictionary.com/word/search.html?q=paradigm>)

Definition: Paradigm Shift

A significant change in the paradigm of any discipline or group.

(<http://dictionary.reference.com/browse/paradigm+shift>)

- Paradigm shifts drastically change the way a subject is approached
- Most (if not all) paradigm shifts encounter resistance from those heavily invested in the old paradigm
- **CDM requires a paradigm shift for information security to allow automation**

Paradigm Shift Example



Spitfire	Cyber-Security
The Spitfire (fighter aircraft) was developed in the run up to WWI	Today, we may be in a similar run up to (or already be in) cyber conflict that could have high economic significance.
The design of the Spitfire as a single winged plane focused on different objectives from earlier aircraft was resisted by the “Air Office” and others used to the earlier Bi-plane design.	There is considerable resistance to moving from key concepts in traditional FISMA and ISCM that may need to change.
One activity that drove the innovation necessary to change design was having a competition to succeed to speed trails (a metric for success).	Correctly done, ISCM can provide such outcome based success measures to evaluate different cyber-operational approaches.
The “Air Ministry” originally rejected the Spitfire because it failed to meet their “Specifications”. However, those specifications were based on a “bi-plane” paradigm of what was needed. The commercial firms building the Spitfire had to publically reject the “Air Ministry” specifications to succeed in building the Spitfire.	To have the best cyber-security, it <u>may be</u> necessary to reconsider some old assumptions about what “compliance issues” are most important to look at in measuring “success”.
The Spitfire was critical to winning the Battle of Britain, which stopped the German advance, and provided a beachhead for Allied victory in Europe.	Will we need to make similar changes to our paradigm of cyber-security (especially in ISCM) in order to win the cyber-conflicts as we move ahead?

Desired State Inventory/Specification

- Many organizations use sensors to collect the actual state of system configurations that affect security.
- Few organizations create an automated desired state specification that can be simply compared by computer with the actual state to filter for defects.
- A key paradigm shift of CDM is to express the desired state specification (or inventory) in data so that it can be easily compared to actual state and find defects.
- It is no longer feasible to do this manually and still find defects fast enough.

Automation

- Many security testers use methods such as interviewing personnel and reviewing documents to assess control compliance.
- Manual methods are both slow and expensive.
- A key paradigm shift of CDM is to automate the discovery of defects.
- It is no longer feasible to do this manually and still find defects fast enough.

Automated Prioritization

- With automated desired state specifications and automated discover of defects, it is possible to check for and find millions of defects.
- If we can't also prioritize these in an automated way, people will get lost in a mass of defects.
- A key paradigm shift of CDM is to use risk scores (discussed later) to automatically prioritize defects.

Fix the Worst Problems First

- Most commercial dashboards count defects, but do not sort defects to list the worst problems first.
- Many commercial dashboards have prioritized data, but do not use it to show the worst problems first.
- The problems need to be presented to the group directly responsible to fix the problem!
- **A key paradigm shift of CDM is to have a dashboard that:**
 - Focuses operational personnel on just the defects they should fix AND
 - Lists the worst problems first

New Roles

- Implementing these paradigm shifts moves the workload and changes roles.
- **Key paradigm role shifts in CDM are to:**
 - Focus operational personnel on fixing the worst problems first, becoming much more efficient
 - Having a few personnel maintain the “desired state” and “scoring rules”
 - Automate as much assessment as possible, to free more staff to focus on better engineering and fixing issues
- **As the paperwork workload goes down, more time can be devoted to engineering and directed repair.**

Technical and Managerial Options

- The implementation training will outline many implementation choices. These include:
 - Input to the technical requirements that might be unique to your organization
 - Working with FNR and the Continuous Monitoring as a Service (CMaaS) Contractor to find an approach to CDM that works for your organization
 - What kinds of sensors will work best
 - How to organize your reports
 - Local scoring and grading
 - Making your own risk acceptance decisions
 - Deciding what to fix first

Performance Impact

- CMaaS contractors are required to operate on the network with minimal network/device performance impact.
- Assuming each D/A has a way to measure that impact, it is possible to determine the load from CDM operations by:
 - Collecting data during normal operations with CDM
 - Collecting data from a random sample of times when CDM is “turned off”
 - Comparing the performance in each case
 - If performance impact is larger than it should be, the CMaaS contractor is responsible to lower that impact to acceptable levels.
- DHS will work with D/As to monitor this for each CMaaS contractor.

Security Impact

- Each CMaaS contractor is required to have a security plan to:
 - Protect CDM data
 - Make sure CDM doesn't weaken network defenses
 - A provisional authorization package will be available for each CMaaS contractor.
- D/As may have additional tests they wish to run before granting authority to operate.

Technology Changes Change Security

- As new technologies arise, the CDM program will :
 - Assess potential requirements for new capabilities, introduced by new attack paths
 - Update diagnostics to test for new controls added by NIST
 - Document architectures and concepts of operation, as needed to keep CDM functioning with new technologies

END OF SECTION

Continuous Diagnostics and Mitigation (CDM)

Module 1 – CDM Overview

Part 4 – Dashboard

Topics

1. Why a Dashboard? Who is it for?
2. Use for prioritizing work (System Admins)
3. Use for evaluating risk management (ISSO, CISO, CIO, etc.)
4. Grades to communicate to senior (mission) managers
5. Normalization of scores
6. Risk thresholds to define risk triggers
7. Architecture
 - The sensors feeding the Base, and the Base feeding the Federal – no Intermediates.
 - The idea of the Base summarizing data and sending it up
 - The idea of containers (with a diagram) and container-level authorization
8. Expected frequency of updates from sensors and to the Federal

Why a Dashboard?

- Each CDM user requires the right information for their particular role:
 - Operators need to know what to fix first, and how to do it.
 - Managers need to know what kinds of problems create the most risk, and whether risk is going down.
 - Business owners need to know if the “IT” staff are managing their risk.
 - Authorizing officials need to know risk is within acceptable limits, and whether it is “jumping” up.
 - OMB, OIG, etc. needs to know security capabilities are improving and increasingly mature – thwarting “attacks”.
 - Existing sensors and their displays are usually too technical and non-integrated to fulfill these needs.
- The CDM dashboard is the best way to do all of this.

Fixing the Worst Problems First

- System Administrators (SAs) need to be able to find the worst problems to fix those first. This might mean:
 - The worst devices on a LAN
 - The defect-types creating the most risk on a LAN
 - The worst defects on a device
- The dashboard should support all these views.
- SAs should be able to glance at the dashboard each day and get a quick to-do list.
- Since the goal is to fix these problems, the dashboard should link to guidance on how to fix these problems. That will help the SAs do their job.

Risk Management

- Local ISSOs need to know how well different groups are doing in managing risk. This means the dashboard needs a way to present average risk per device (for example) to fairly compare groups that manage a few or many devices.
- Business managers need a simple way to know when security issues demand their attention. Letter grades (or some equivalent) is an easy (and non-technical way) to communicate that something more is needed (priority, resources, training).
- Agencies need a way to define their own risk management parameters AND there needs to be a way to understand the risk across the entire Federal government.
- **An effective dashboard must meet a wide range of needs.**

Management Needs Risk Triggers

- With billions of checks, management cannot focus on every defect.
- The dashboard needs an automated way to flag significant patterns, or “triggers*.”
- Examples include the following:
 - Risk rises above a control limit
 - Across the LAN
 - On a specific device
 - Risk is steadily rising for N days
 - Risk jumped more than X in one day/week/etc.
- When triggers are passed, risk managers need to look at the situation in a way proportional to the total risk involved. Several agencies have existing models for how to organize accordingly.

* Grades are a simplified version of triggers.

Dashboard Architecture 1



The diagram illustrates a three-tier dashboard architecture. At the top is a single blue box labeled 'Federal Dashboard'. Below it is a stack of five blue boxes, with the bottom-most one labeled 'Local Dashboards'. At the bottom is a stack of five green boxes, with the bottom-most one labeled 'Capability Collection Sub-Systems'. The boxes are arranged in a staggered, overlapping fashion to show a hierarchical relationship.

Federal Dashboard

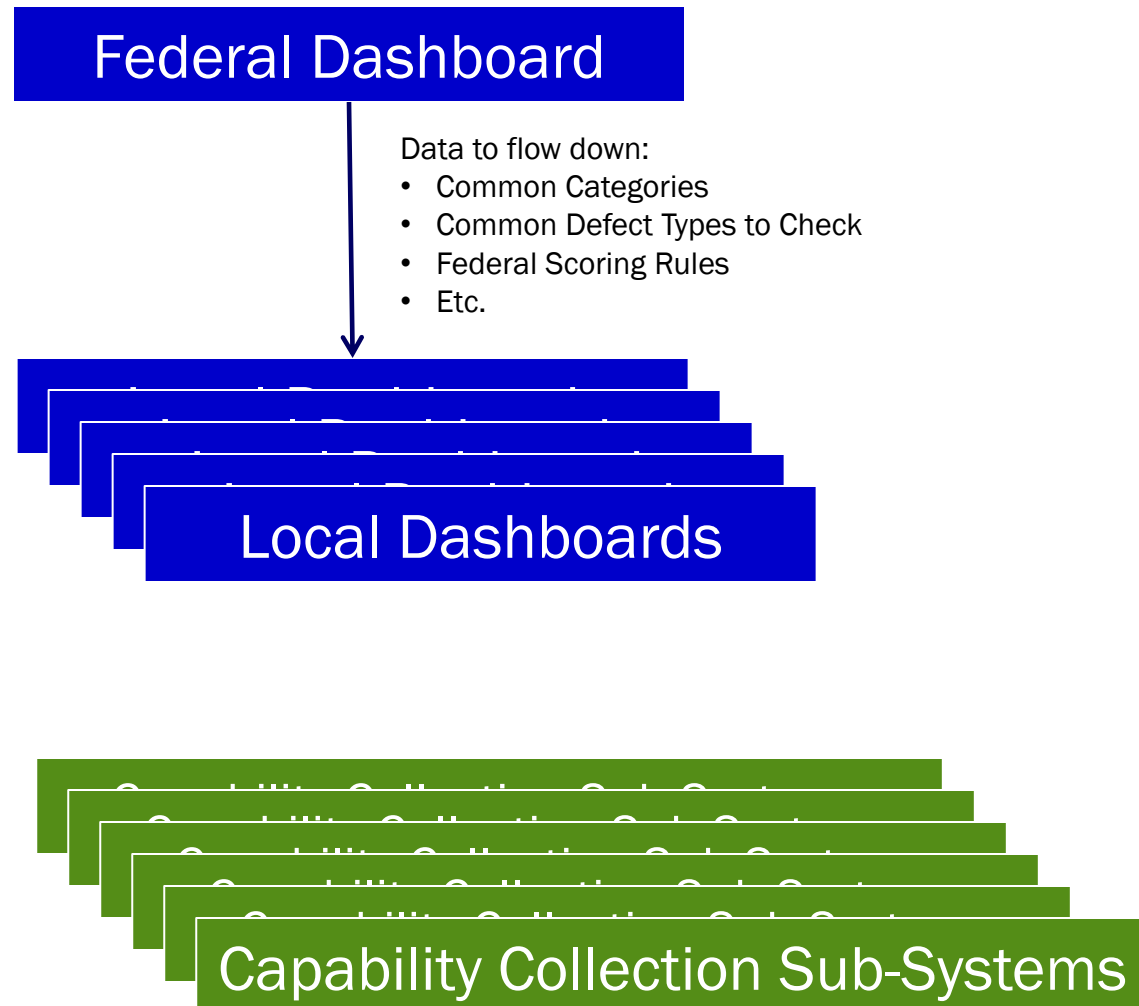
There may be several federal dashboards, including for classified and unclassified networks.

Local Dashboards

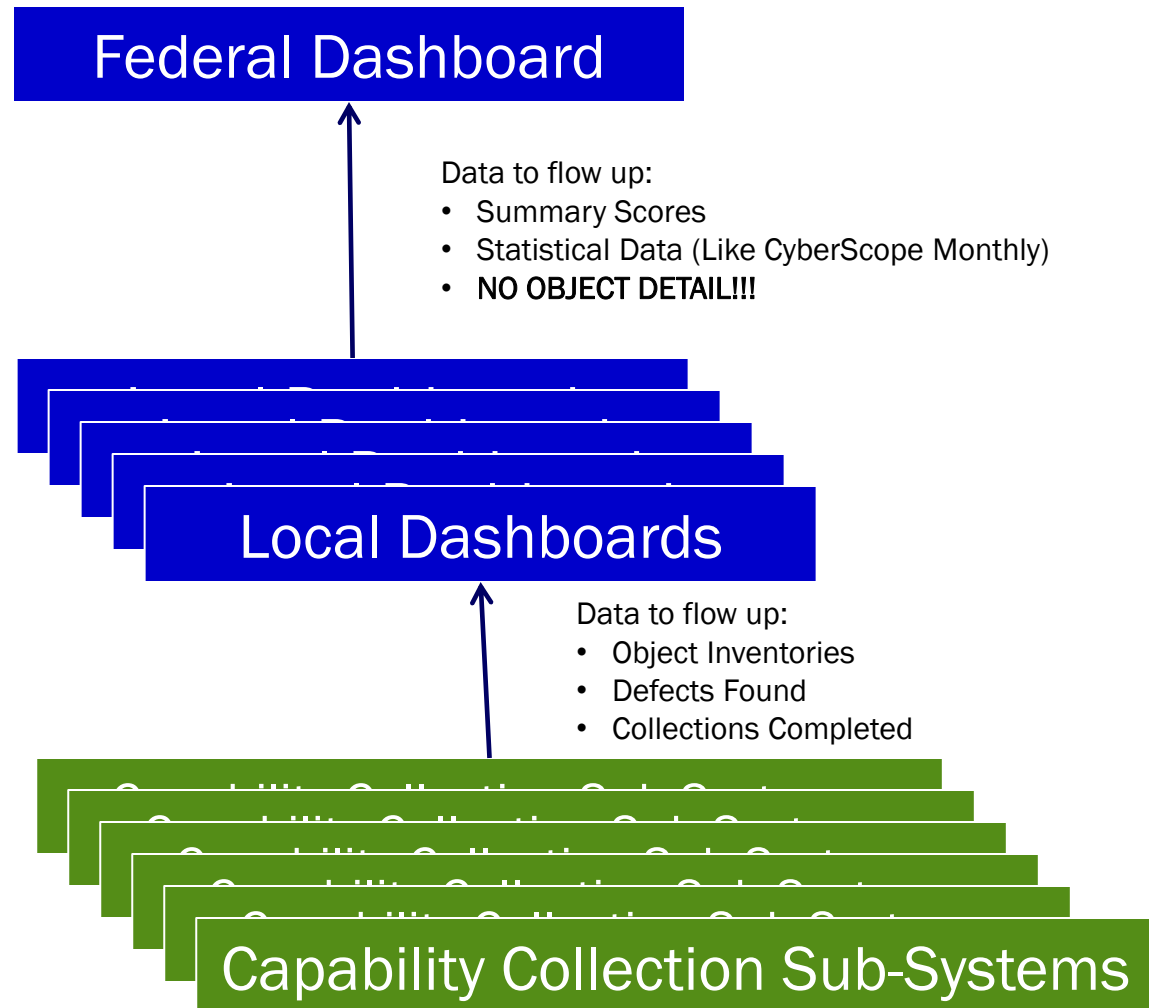
Ideally there would be one of these for each large agency (or equivalent). Security would limit staff to viewing only issues within their purview.

Capability Collection Sub-Systems

Dashboard Architecture 2



Dashboard Architecture 3



Dashboard Architecture 4

- The dashboards will define a set of hierarchical groups by which object-risk can be summarized and reported. Examples are given in each capability, but common categories include:

Group by	Business Purpose
Device manager	To send problems/defects to be fixed.
Device role	To be able to specify authorized SW, settings, and patch requirements by role.
Business owner	To inform the owner about risk to their business functions.
Organization supporting the device	To establish service-level agreements and measure performance
Applications supported	To allow applications to inherent controls from network, as appropriate.
Common Platform Enumeration (CPE) or equivalent	To allow reporting by device product/vendor etc.

Dashboard Update Frequency

- The CMaaS contractor is responsible to collect data from sensors and update dashboard inventory and defect data.
- This should happen at least three times a day (every 8 hours).
- Not all data will be collected every eight hours.
 - All data will be collected every three days
 - The update requirement is to refresh data that changed in the last eight hours.
- Given that most operations will look at the dashboard daily, and fix the worst problems first, this schedule ensures that the most current data is generally available each “shift.”

END OF SECTION

Continuous Diagnostics and Mitigation (CDM)

Module 1 – CDM Overview

Part 5 – Scoring and Grading

Topics:

1. Why Score?
2. Adding Scores
3. Why Grade?
4. Getting Comparable Scores
5. Generating Friendly Competition
6. Federal/Local Scores
7. Considering Threat and Impact
8. Example of scoring and Aging
9. Limits
10. Why Maturity Metrics?

Why Score?

- CDM will typically find millions of defects in a large network-more than can be fixed in a short time.
- **It is essential to focus on the worst problems first.**
- Risk scores are a way to prioritize the work, so that:
 - Each team can be given a prioritized to-do list,
 - The focus is on the worst problems today.
- Because of the typically large number of defects, this prioritization is difficult to do manually.

Adding Scores

- To add scores, the individual scores must be on a ratio scale.
- **Counting Defects:**
 - Many dashboards count the number of defects.
 - This is a weak measure of risk, because not all defects are associated with the same risk.
 - **Counting assumes each defect has a risk of 1**, implicitly.
- **Ranking Defects:**
 - Defects are often ranked high, moderate and low. We might then assign a 3 to high, 2 to moderate, and 1 to low.
 - This is called an ordinal scale because it lets us sort the risks into a rough order of priority.
 - It is impossible to know if a high is really three times as bad as a low. So, adding the scores still doesn't provide a meaningful overall risk.
- **Scoring Defects:**
 - To make adding scores meaningful, we need a “yardstick” that provides a scale of risk.
 - For example, if 10 points equals the worst CVE from the National Vulnerability Database on a single machine, and if other risk scores represent how bad the risk is compared to that yardstick, then adding scores makes sense.
 - This is called a ratio scale, because the ratio between two scores lets you compute how much worse one risk is than the other.

Why Grade?

- Non-technical business managers cannot be expected to understand the significance of a particular 800-53 control not being met.
- But business managers do need to know:
 - When to focus more priority/effort on security (or not)
 - When to provide more training
 - When to provide more resources
- There needs to be a simple way to communicate when security problems are urgent to such managers.
- **Letter grades provides one way to tell business managers when their information security team needs more support.**
- Others methods include triggers and stop-light charts.
- Grading seems to have the most impact.
- It will only work if the grades are trusted to be fair and transparent.

Getting Comparable Scores

(Normalization)

- Grading requires a way to compare groups of different sizes
- To get the risk on any one object (say a device) scores will be added from all defect types:
 - For this to be meaningful, the scores must be on a “ratio” scale – the relative size of the score should reflect the relative amount of risk.
- Scores can then be added across all objects in a group to get a total risk score.
 - A group might be all objects managed by one team.
 - This score is highly affected by the size of the groups.
 - More objects probably means more risk.
- To take out the effect of object size for the group we generally divide total risk by the number of objects to get average risk per object.

Generating Friendly Competition

- A primary benefit of scoring and grading is that people can easily compare their security performance to others.
- Experience has shown that this can create friendly competition that lifts all boats.
 - Poor performers start to improve, perhaps asking better performers for help.
 - The better performers then improve to stay ahead in a self-reinforcing cycle.
- But that only works if key conditions are met:
 - The scores are fair and transparent
 - Management uses Theory Y
- These critical factors are discussed on the next two slides.

Fair and Transparent Scores

- The natural initial assumption about scoring is that it is not really to help line security workers, but is a stick for punishment.
- If that view is true, scoring will not improve performance.
- Evidence indicates that participants will then ignore the dashboard.
- To be viewed as fair and transparent, the dashboard and scoring system must do (at least) the following:
 - Assign defects to those actually able to fix them (not someone else).
 - Have a low rate of false positives.
 - Tell people what specific defects to fix to raise their score.
 - Tell people which fixes raise their scores most.
 - Tell people how to make the fix.
- **People need a way to resolve issues when they think the system is unfair.**

(See the human factors section for more on this key topic.)

Local and Federal Scores

- **Federal scores** are designed:
 - To score things that are clearly a problem across federal networks
 - To let organizations compare their security posture to others
 - To decide whether to share information
- D/As may also create **local scores** to manage their own risk. This has the following features:
 - Ability to add extra defect checks (and scoring) to deal with D/A specific issues and risk tolerance
 - Changing local scores to be higher or lower than federal scores for federal defects, to allow work to be prioritized correctly
- By having two systems, CDM can compare D/As while ensuring that each D/A has full authority and tools to manage its own risk.

Scoring Concepts

- CDM scoring includes the following concepts:
 - Base Score (Vulnerability)
 - Threat Factors
 - Impact Factors
 - Grace Period
 - Increasing Risk Over Time (aging)
 - Risk Limits
- Assumption: $\text{Risk} = f(\text{Threat}, \text{Impact}, \text{Vulnerability})$
- These concepts are discussed in the next few slides...

Scoring Concepts: Base Score

- The basis for scoring is called the base score.
- The base score should express **how vulnerable (easy to attack)** the defect makes the system.
- Example: The CVSS (Common Vulnerability Scoring System) scores for CVEs in the National Vulnerability Database (NVD) are Vulnerability Scores.
- The base score expresses **VULNERABILITY**

Scoring Concepts: Threat Factors

- Threat factors express how likely a vulnerability is to be exploited.
- Threat may come from several directions. For example:
 - A particular weakness may be the target of a common active attack tool. This may change over time.
 - A particular location, person, machine, etc. may be more frequently targeted for various reasons.
- A threat factor is a multiplier expressing how much above a nominal base level is the likelihood of exploitation.
- A specific defect may have several factors that increase threat. For example:
 - A defect targeted by actively used tools (e.g., three times as often as most defects) on a device used heavily by a person targeted more often than most people (for example, due to job role).
- Because multiple threat factors compound risk, the organization may want to place a limit on the overall threat multiplier.
 - For example, you might specify that it is not over five.

Scoring Concepts: Impact Factors

- Impact factors indicate how much more impact a successful attack on a specific object would cause, compared to a nominal baseline object.
- For example:
 - An attack on a normal user's Outlook (local mailbox) would be less impactful than an attack on one of an organization's main Exchange (e-mail) servers.
 - How much more? That is the Impact factor.
- A given defect may have multiple impact factors. For example:
 - One based on who (or what business function) uses a device, and
 - One based on the data sensitivity of stored on the device
- Maxima may also be needed.
- **Downstream impact is a key concept.** While a device itself may be low impact if compromised, its impact score probably should consider what other devices can be compromised if it is used as an attack platform, and the essential services or sensitive information that are dependent on those devices.

Basic Scoring Formula

- The basic Scoring Formula is:
$$(TF1 * \dots * TF_n) * (IF1 \dots * IF_n) * \text{Base Score}$$
- Where:
 - TF_x = Threat Factor x
 - IF_x = Impact Factor x
- To simplify we ignore the impact of limits on the threat factors.
- There may also be a maximum on the total score.

Scoring Concepts: Grace Period

- A grace period can be applied.
- This assumes people need a short time from the detection of a defect to respond. For example, maybe you assume that Anti-Virus updates only need to be done once a week, so you assign a grace period of 7 days from the time of the last update before scoring starts.

Grace periods help provide a sense of fairness.

Scoring Concepts: Risk Aging

- Some risks may increase over time.
- In this case, the risk score should go up over time.
- For example, the longer an anti-virus system is not updated, the higher the risk, because more and more definitions are being missed.
- “Aging” a score is a way of requiring it to increase over time to reflect the increased risk.
- Without a maximum, aging will eventually increase the score so much it dominates all others. If that is not correct, you can specify a maximum overall score to limit the effects of aging.

Scoring Concepts: Risk Limits

- As noted above, maxima (or risk limits) may be placed on
 - Overall Threat Factors
 - Overall Impact Factors
 - Overall Defect Score
- This is keep single scores from dominating all others, if the combined effect of these escalators is too much.

What Scoring Does Not Measure

- Scoring works best helping line security workers fix the worst problems first.
- It helps with evaluating overall security.
- Scoring does not measure:
 - How capable an organization is to manage risk
 - How likely an organization is to miss defects (given its CDM program)
 - What the level of residual compromise is likely to be
- Maturity metrics are used to measure these higher level concepts.

END OF SECTION

Continuous Diagnostics and Mitigation (CDM)



Module 1 – CDM Overview
Part 6 – Maturity Metrics

Topics

1. **Paradigm Shifts**
2. **What are Maturity Levels and Metrics?**
3. **What is the role of Maturity Metrics in CDM?**

Paradigm Shifts

Maturity Model

- The objective of the CDM Maturity Model is to allow any organization to show the extent of progress toward a robust CDM program, regardless of where they start.
- The CDM Maturity Model:
 1. Has four “Levels” of maturity
 -  2. Applies to each capability, not just overall security, so organizations can implement CDM one capability at a time
 -  3. Measures within each maturity level show how much progress has been made
- These features were added to enable organizations to show incremental progress.

Measuring More than Risk

- CDM uses “Risk Scores” to help prioritize the mitigation of particular flaws and defects, thereby informing measurement of risk management within an organization.
- This works well within an organization, because each organization tends to measure the same defect types.
- However, risk scores are not good measures to compare risk across organizations where the defects evaluated and the quality of the data is different.
- **Maturity Metrics**
 - Measure the quality of the data, so it can be presumed valid and trusted
 - Estimate the amount of residual impact (not just risk) so management can make better risk acceptance/mitigation trade-off decisions
- Maturity metrics therefore enable valid comparisons among different organizations.

Why We Need “Maturity Metrics”

- To measure the effectiveness of a security capability as deployed in the operational environment
- To indirectly test security controls in support of ongoing authorization activities
- To allow one operational solution to align 800-53, 800-37, 800-39, and 800-137
- To evaluate the efficiency of risk management for each local environment
- To evaluate risk management effectiveness independent of changes in technology and innovation

Why We Need “Maturity Metrics”

(Continued)

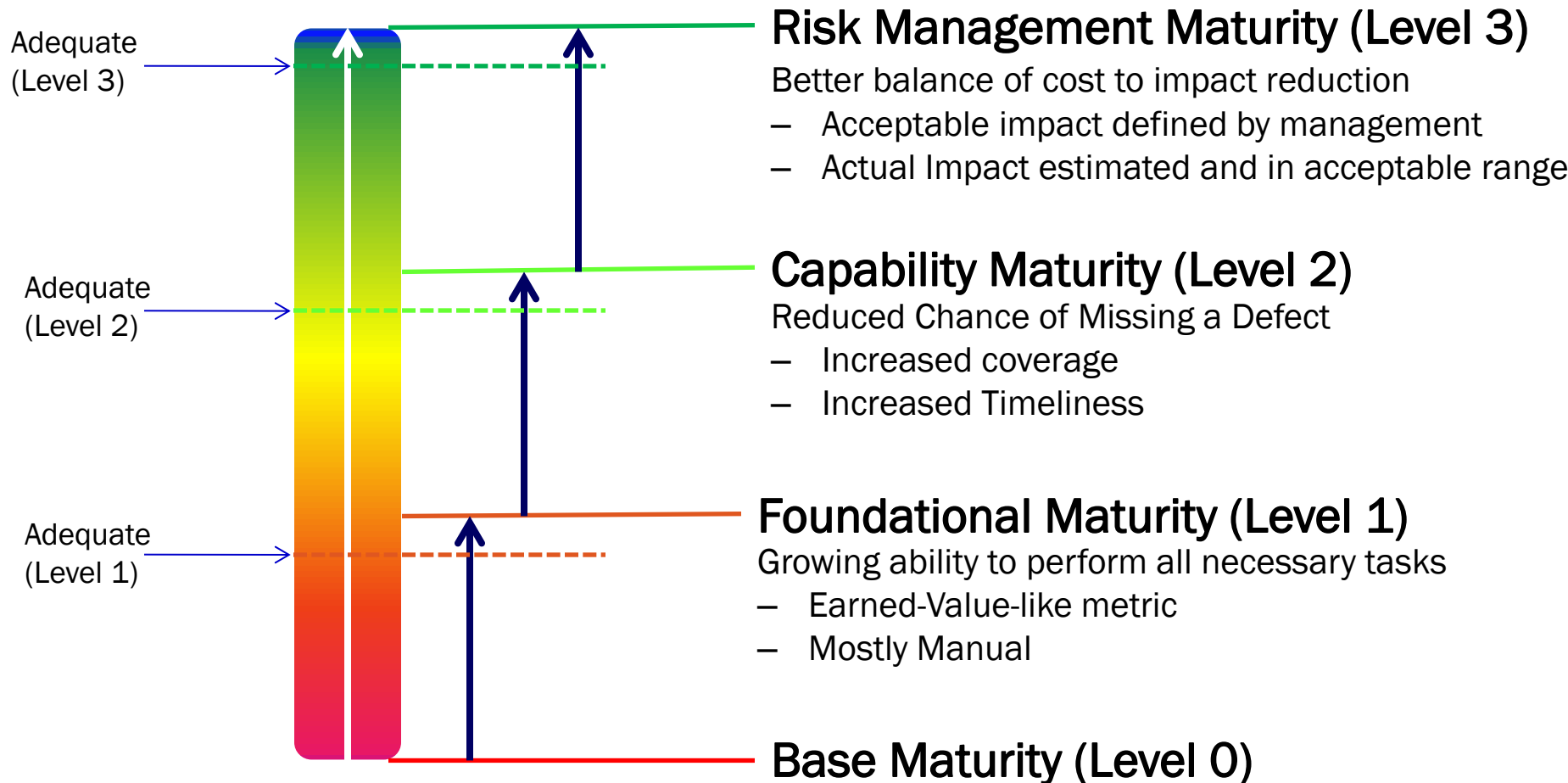
- How does measuring maturity meet these needs?
 - Allows the Federal Government to mandate the required minimum effectiveness of a capability
 - Enables each D/A to deploy optimal solutions for their enterprise
 - Demonstrates incremental improvement throughout the implementation process
 - Objectively evaluates the end result of implementation

What Are the Maturity Levels and Metrics?

CDM Maturity Levels

The standard is adequate performance; not perfection!

Why? The last few % improvement doubles the cost.



Base Maturity (Level 0)

Criteria:

For the selected capability, the organization has demonstrated no ability to perform the steps of the capability for any part of the organization.

Foundational Maturity (Level 1)

Exit Criteria:

- For the selected capability, the organization has demonstrated the ability to perform the steps of the capability for enough of the organization to have begun to implement the capability.
- How fast it does this is not a factor.
- This includes being able to:
 - Define desired states
 - Collect actual states/behaviors
 - Compare the two to identify defects
 - Prioritize the defects
 - Present the results as to-do lists for those actually responsible to respond to the defects
 - Respond
 - Knowing the universe to be covered
- “Enough of the organization” depends on the size of the organization.
 - For very large organizations (>100,000), it’s roughly 20-25% of the organization.
 - For very small organizations (< 500), it’s most efficiently defined as the whole organization.
- Level 1 shows initial progress toward implementation.

Foundational Metric

- The foundational metric is similar to an Earned Value Measure designed to estimate what percentage of the work needed to reach foundational maturity is completed.
- It is measured using a survey, because until this level is reached data is not sufficient to provide an automated metric.
- These surveys have now been used on multiple federal civilian agencies to measure baseline state against which to measure progress.

Capability Metric

- The capability metric can be measured via the CDM dashboard as a by-product of CDM operation.
- It is typically a weighted average of:
 - The percentage of the organization/system “covered” for all selected defect types.
 - How frequently actual state data is collected.
- This assumes the desired state data is complete and current as well.
 - This metric corresponds to the probability of missing a defect, as defects cannot be found before exploited if:
 - Testing is sufficiently complete to identify all defects before exploitation
 - Testing occurs more rapidly than the approximate exploitation tempo of a potential attacker.
- If either completeness or timeliness are too low, the probability of missing defects goes up.

Capability Maturity (Level 2)

Exit Criteria:

- For the selected capability, the organization has a low probability of missing a significant defect for long enough to allow significant risk.
- Typically this metric is based on the ability to perform all of the CDM steps...
 - For the entire organization/system
 - Frequently enough to find/fix defects faster than an attacker can
- Level 2 provides an organization with insight into how much to trust its CDM data.

Risk Management Maturity (Level 3)

Exit Criteria:

- For the selected capability, the organization uses the capability to:
 - Describe the expected impact of residual risk,
 - Uses this information to make budget/resource decisions to reduce and/or accept risk.
- Typically this metric is based on the ability to perform:
 - Residual Risk Estimation (largely automated through CDM)
 - Make sound risk management decisions based on that data.
- Level 3 determines how well the organization manages risk.

Risk Management Metric

- The risk management metric is an estimate of the number of “objects” (for example, devices) typically compromised, given:
 - The parameters of organizational capability metrics
 - How fast defects are fixed once detected
 - How frequently defects “occur”
 - How fast attackers will typically find and exploit specific types of defect
- A risk management metric might say that given current HWAM capability, there are typically 20 compromised devices on the network and that this is 85% of what management has determined is an acceptable risk.
- Metrics at or below 100% are acceptable.
- Metrics above 100% are not.

How Do You Know if Risk is Effectively Measured?

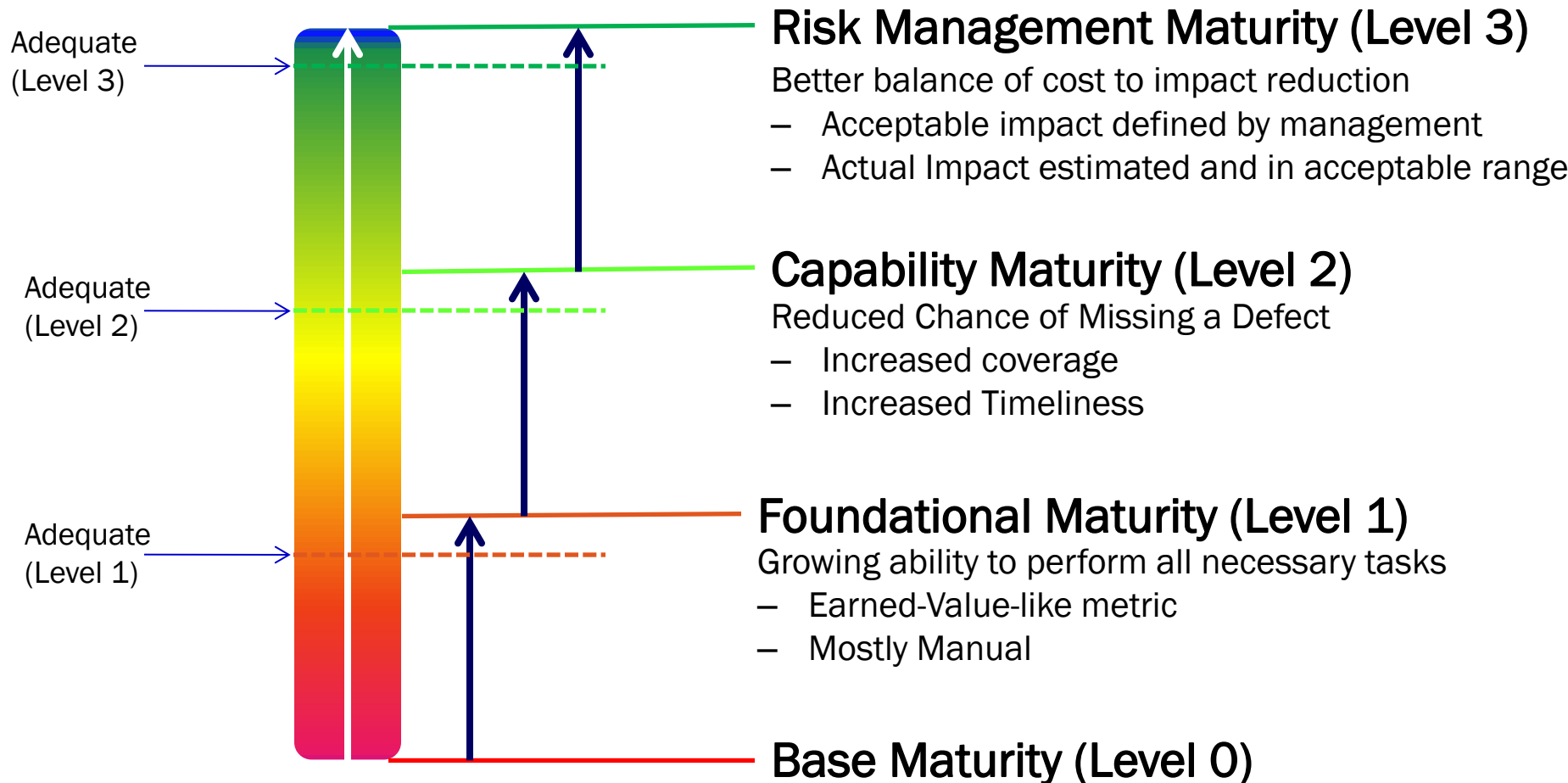
Based on

- Known attack scenarios
- Known threats/impacts
- Real-world incident data can be used to validate that being more mature equates to more effective capability.

CDM Maturity Levels

The standard is adequate performance; not perfection!

Why? The last few % improvement doubles the cost.



What is the Role of Maturity Metrics in the CDM Model?

Where Do Maturity Metrics Fit?

- The CDM Wheel determines what high level results (capabilities) need to be covered.
- The mapping of 800-53 controls to the “Wheel” show how each capability can be implemented.
- Risk scores prioritize defects and provide coherent performance metrics.
- Root cause analysis identifies what to systematically improve through better engineering.
- Both Risk Scoring and Maturity Metrics inform where to invest to have the most impact. Both are objective measures.
- Maturity metrics provide:
 - When data is “good enough” (RMF/assessment management)
 - When the residual impact is low enough (risk management)

**These parts are all complimentary and work together.
(See the next diagram)**

REALIZING THE RMF

ARCHITECT/DESIGN/BUILD

Most cost effective place in system lifecycle to improve security capabilities

CATEGORIZE SELECT IMPLEMENT ASSESS AUTHORIZE

WHAT TO IMPLEMENT



Determine what controls are deficient and identify alternative implementation



WHAT TO IMPROVE

Systemic Risk Issues

OPERATE

Only cost effective to test operational effectiveness in the operational environment

MONITOR

WHAT TO TEST



Desired State | Actual State
ACTION THE DIFFERENCE



WHAT TO FIX

CSIS: 20 Critical Security Controls

The CSIS 20 Critical Security Controls are the high priority actions that when implemented, mitigate current risk.

The CSIS 20 Critical Security Controls are represented by the capabilities within the wheel. Ex: Mapping for Network/Physical Access Control and Configuration Settings



MANAGE ASSETS

- Network/Physical Access Control
 - CC7: Wireless Device Control
 - CC10: Secure Network Device Configuration
 - CC11: Limitation/Control of Network Ports, Protocols, and Services
 - CC13: Boundary Defense
- Configuration Settings
 - CC3: Secure HW/SW Configuration
 - CC5: Malware Defenses
 - CC7: Wireless Device Control
 - CC10: Secure Network Device Configuration

The CSIS 20 Critical Security Controls can be used to prioritize what controls/capabilities to implement/measure first.

Demonstrating Maturity of Operational Management Capabilities (the Wheel) shows you can manage risk.



Risk Scoring helps you manage your risk.



Security investments:
- Increase Maturity and/or
- Reduce Risk Score
Both are measurable.

Effective Security Control Implementation = Mature Operational Management Capabilities = Ability to Manage Risk

END OF SECTION

Continuous Diagnostics and Mitigation (CDM)

**Module 1 – CDM Overview
Part 7 – Ongoing Assessment**

Topics

1. Paradigm Shift
2. Mapping 800-53 Controls to Capabilities/Attack-Types
3. Defect Type Tables
4. Control Allocation Tables
5. Use Case (Sample Process)

Paradigm Shift

Paradigm Shifts



Old Paradigm	New Paradigm
Some think the NIST assessment methods require use of interviews and examinations	NIST says that tests may always be substituted for interviews and examinations, since tests are more rigorous methods.
Manual Tests	The CDM concept of operations is AUTOMATION of the NIST “Test” assessment method.
We can only automate technical control assessments.	We can automate most technical AND non-technical control assessments.
We should focus on testing controls.	We should focus on testing the results controls are intended to achieve.
We must directly test each control.	We can indirectly test whether a control works (and whether it matters) as long as we test overall results.
When we find a failed test, we just fix what failed.	When something fails, we need to find the root cause, so we don’t just “treat the symptoms”. This leads to better engineering.

Interview-Examine-Test

- NIST defined three Assessment Methods in 800-53A Appendix D:
 - Interview
 - Examine
 - Test
- In general **“Test” is the most reliable method** to find defects, though the others may be important in root cause analysis.
 - NIST allowed non-Test methods primarily in cases where NIST felt TEST might be onerous.
- **NIST assessment method guidelines are optional, not mandatory.**
- **CDM focuses on using the more rigorous test method to find as many types of security defects as possible.**

Automating the TEST Method

- NIST 800-53A defines the Test Method as follows:
 - The process of exercising one or more assessment objects under specified conditions: ***to compare actual with expected behavior, the results of which are used to support the determination of security control existence, functionality, correctness, completeness, and potential for improvement over time.***
- This is the basis for the overall CDM concept of operations:

NIST TEST	CDM Concept of Operations
The process of exercising one or more assessment objects (to know expected behavior)	<u>Collect the Desired State</u> , which is often a baseline configuration in DATA.
The process of exercising one or more assessment objects (to know actual behavior)	<u>Collect the Actual State</u> (which is often called an inventory) in DATA
To compare actual with expected behavior	<u>To AUTOMATICALLY compare actual with expected behavior</u> and to AUTOMATICALLY prioritize defects.

How Can We Automate Tests of Non-Technical Controls?

- A non-technical control might be ensuring that the organization knows the manager for each device [CM-08 (4)].
- To automatically test this:
 1. Require the desired state specification for the device to specify an individual or group to manage the device
 2. Verify (in actual state) that:
 - The manager is specified for the device
 - The manager is still an active person or group
 3. Test that the number of devices assigned to that person or group is not an unreasonably high number for them to manage
- Once we automate the desired state specification, the rest can be easily computed.
- The key to automation of testing is having an automated desired state specification that can be easily compared to the actual state.

Testing More than Controls (Results)

- In 800-53 (rev 4), NIST introduced “Security Capabilities” as: A collection (set) of security controls that work together to achieve an overall security purpose (NIST 800-53 Rev4, p. 21.)
- CDM organizes controls into such groups.
- It will often be useful to test whether the overall security purpose of a security capability is being met
 - As NIST notes in 800-53 rev4, p. 21
 - Failure of an individual control(s) may not affect this overall purpose
- In such cases, the individual control(s) may be judged less important.
- **The key is to know if the broad purpose is being achieved.**

Indirect Testing

- If it is accepted that failure of an individual control(s) may not affect the achievement of the overall purpose...
 - ...And it is possible to measure that achievement...
 - ...The individual control(s) may be judged less important
- In this case, it may be useful to accept a fourth assessment method which CDM calls “Indirect Testing”
 - Indirect Testing means testing whether the result is achieved, and IF IT IS, accepting that the controls necessary to achieve that result must also be working (without testing each).

Root Cause Analysis

- Whether using direct or indirect testing, root cause analysis may be necessary.
- This is not spelled out in the NIST methodology, but recognized by NIST as essential.
- Root Cause Analysis (RCA) is a method of problem solving that tries to identify the initial (root) causes of faults or problems
 - So that problems can be addressed at their source.
 - RCA avoids “treating the symptoms.”
- RCA may be required when:
 - The same things keep breaking because the source problem is not being fixed
 - Indirect testing finds things aren’t working, because indirect testing doesn’t point to a specific control to fix.

Assessing Risk

- An assessment method is required that more accurately assesses risk.
- NIST notes that knowing whether the capabilities (as a whole) are being achieved is key to better risk assessment:

Ultimately, authorization decisions (i.e., risk acceptance decisions) are made based on the degree to which the desired security capabilities have been effectively achieved and are meeting the security requirements defined by an organization. These risk-based decisions are directly related to organizational risk tolerance that is defined as part of an organization's risk management strategy. (800-53 rev4, p. 21)

- The CDM focus on security capabilities supports better risk assessment.

Mapping Controls to Capabilities

Why Automated Mapping

- Initially, DHS used a manual process to map CDM to NIST 800-53 rev3
- This demonstrated that an automated process is required
- The automated process DHS developed uses “regular expressions” which looks for key word and/or phrase combinations that map a control to a capability.
- These were validated with the 800-53 leads at NIST until they produced reliable results.
- About 200 expressions are used to map the controls to the capabilities.

Why Map Control Items (NOT Controls)?

- Each 800-53 controls may have many parts with:
 - Sub-sections like a), b), c), d) ...
 - Numerous “enhancements”
 - Enhancement sub-sections like a), b), c), d) ...
- The initial manual mapping suggested that the parts of a control do not always map to the same security capability (purpose/result).
- So, DHS mapped the individual control items, rather than the full controls to get better resolution/precision.

Example of “Regular Expressions”

- To find controls related to hardware asset management, DHS used the regular expressions like the following single example:
 - Any control item that maps to HWAM if it contains “inventory”
- Sometimes the regular expression was more complex. For example:
 - Any control item that maps to HWAM IF it contains “supply chain” and NOT “monitoring”
- Checks were not case sensitive.

False Positives/Negatives

- No such automated method can be perfect.
- Generally such tests have two types of errors
 - False negatives (correct mappings that it missed)
 - False positives (extra incorrect mappings)
- It is accepted that as one type of error is reduced, the other type increases.
- **DHS attempted to minimize false negatives (missed mappings).**
 - This was because if it missed anything, manually checks would be required to find them.
 - DHS then accepted that it would be most appropriate to manually weed out remaining false positives.

All Rules Used to Find HWAM-Related Controls

Regular Expressions for HWAM

unsupport AND *system*

thin nodes

tamper resistance

supply chain and not *monitoring*

property

mobile and *locations* and *risk*

inventory

Manually marked as HWAM

heterogen

function isolation

collaborative computing device

change control

personally owned OR *non-organizationally owned systems*

baseline and *config*

anti-counterfeit

- Similar expressions were developed for each capability.
- Each new version of 800-53 requires minor adjustments of the rules where new lexicon is used. This is still must easier and more reliable than manual mapping.
- Sometimes the same expression works for several capabilities. For example, baseline configurations apply to both HWAM and SWAM.

Mapping Controls to Capabilities

- This material was presented to demonstrate how capabilities are related to the C&A process.
- This is not work that each D/A has to repeat. It's done once by the CDM program, and used by each D/A.

No D/A action is required.

Defect Type Tables

CDM Defines Defect Types

- Defect Types are the automated tests to be used to assess the extent to which a capability is met.
- Each defect type has
 - An ID Code
 - A Name for easy identification
 - A test to determine if the defect is present
 - Options of how to address a defect
 - A decision to implement the test (or not)
- The defect types can be expressed in a table.

Core and Non-Core Defect Types

- For each capability there are typically:
 - Core (federally selected) defect types that assess the overall resilience of the capability, and
 - Non-Core (optional, local) defect types that primarily help with root cause analysis related to individual controls
- Below these two groups of defect types are coded as follows:
 - CAP - Xn where:
 - CAP is the capability code
 - X is either
 - F for “Federal” (i.e., Core Defect Type)
 - L for “Local” (i.e., Non-Core Defect Types)
 - N is a sequence number within CAP-X
- So HWAM-F2 is the second (2) Core (F=Federal) defect type for the Hardware Asset Management (HWAM) capability.

HWAM Has 2 Core Cyber Defect Types

ID	Name	Test	Mitigation Options
HWAM-F1	Unauthorized Devices	Device in the Actual State but not in the Desired State Specification (Baseline Configuration)	<ul style="list-style-type: none">• Remove Device• Authorize Device OR• Accept Risk
HWAM-F2	Unmanaged Devices	Device In Actual State and in Desired State but no “appropriate” manager assigned	<ul style="list-style-type: none">• Remove Device• Assign Device OR• Accept Risk

- These defect types assure that any unauthorized or unmanaged devices are flagged as risk, and action to resolve this is prioritized.
- This addresses the core purpose of this capability.

HWAM Has 2 Core Data Validity Defect Types

ID	Name	Test	Mitigation Options
HWAM-F3	Non-Reporting Devices	In Desired State but not in Actual State	<ul style="list-style-type: none">• Restore Device Reporting• Declare Device Missing OR• Accept Risk
HWAM-F5	Non-Reporting Defect Types	Defect Types are selected, but the HWAM Actual State Collection Manager does not report testing on all devices	<ul style="list-style-type: none">• Restore Defect Type Reporting• De-Select Defect Type• Accept Risk

- These defect types assure that if there are problems with the actual state data, it will be flagged and prioritized for mitigation.
- This addresses whether to trust the data on the core purpose of this capability.
- Maturity metrics also measure whether data is complete and timely enough.

Non-Core Defect Types

- After creating the core defects types defined above, one can review the control items mapped to HWAM to find what additional defect types can be added to cover all low-moderate-high baseline control items.
- These can be selected for implementation based on the impact level of the system or the owner's (or D/A's) risk tolerance.
- These non-core tests tend to related to some detailed aspect of the core defect types.

HWAM Has 9 Non-Core Defect Types (Part 1)

ID	Name	Test	Mitigation Options
HWAM-L1	Device for Travel	Device type or subcomponents do not meet D/A defined rules (before or after travel). Only applies to HWAM if certain device types and/or sub-component configs are approved for travel.	<ul style="list-style-type: none"> Remove Authorization to use for travel Correct configuration Accept Risk
HWAM-L2	Unauthorized Device	Device must be in the desired state and subsequently approved by a separate authorized person from the person who added it and manages it.	[Same as F1]
HWAM-L3	Required device not installed	Device in desired state, authorized, and has not appeared in the actual state after [an organization-defined] number of collections.	<ul style="list-style-type: none"> Install device Remove requirement Accept Risk
HWAM-L4	Unapproved device owner	The device owner is other than a value in an approved list. (Could also apply to sub-components.)	<ul style="list-style-type: none"> Remove Device Correct Ownership Accept Risk
HWAM-L5	Unapproved Supplier or Manufacturer	The device supplier or manufacturer is not in an approved list	<ul style="list-style-type: none"> Remove Device Correct Supply Chain Data Accept Risk

HWAM Has 9 Non-Core Defect Types (Part 2)

ID	Name	Test	Mitigation Options
L6	Subcomponents not Authorized. All controls on hardware configuration.	Subcomponents added to the actual and desired state, and system verifies that [organization-defined sub-component types] are authorized or creates a defect	<ul style="list-style-type: none">• Remove Sub-Component• Correct Configuration• Accept Risk
L7	Authorization reached Sunset	Track an authorization sunset date, which can be expired by trigger events. Score all devices past their sunset date as unapproved.	<ul style="list-style-type: none">• Re-authorize• Remove Device• Accept Risk
L8	Required Device Data	Track additional device data and score devices that don't have that data	<ul style="list-style-type: none">• Add Data• Remove Device• Accept Risk
L9	Proposed Changes too old	Proposed changes not approved after [organization-defined timeframe]. Assumes L2 is selected.	<ul style="list-style-type: none">• Withdraw proposed change• Approve proposed change• Accept Risk

D/A Action on Defect Types

- The CDM program has identified the defect types needed to cover assessment of all 800-53 controls.
(Subject to NIST review.)
- D/As can clearly define other defect types if they feel they are needed.
- It is hoped that this will either not be required, or that these can be shared with other D/A.
- D/As will select which defect types are needed for each system. See the use case, below, for more information.

Control Allocation Tables

Control Allocation Tables (CAT)

- Control allocation tables were proposed by the DHS CISO in an initial pilot of continuous assessment. The CDM program finds them very useful and has adopted them for the CDM program.
- For each control they show **within** a **capability**, they show how that control can be tested.
- Examples follow...

HWAM Low Baseline

Control Allocation Table - CAT

Control Item	Applicability	Inherited by	Diagnostic Responsibility	Defect Metrics	Mitigation Responsibility	Selected	Risk Acceptance	Frequency	Impact
AC-19f	Network	Applications	CDM Check	L1 & L6	NetOps			< 4 days	
AC-19g	Network	Applications	CDM Check	L1 & L6	NetOps			< 4 days	
CM-02	Network	Applications	CDM Check	F1, L7 and L3	NetOps			< 4 days	
CM-08a-1	Network	Applications	CDM Check and CDM Maturity Metric	F3 and Level 2 Maturity or better	HWAM Operator			< 4 days	
CM-08a-2	Applications	n/a	Automated & Manual IV&V	IV&V shows boundary is maintained	Application ISSO			D/A determined	
CM-08b	Network	Applications	CDM Check and CDM Maturity Metric	F3 and Level 2 Maturity or better; or below Level 2 but metric shows timeliness is adequate.	HWAM Operator			< 4 days	
CM-08 (4)	Network	Applications	CDM Check	F2	DSM			< 4 days	

Many columns may be customized for each D/A -- especially inheritance, responsibility, defect types, and the blank columns. However, the intent is to provide a framework to support control selection and assessment planning with minimal customization.

Control Allocation Table Columns

Column	Meaning
Control Item	Maps back to the part of an 800-53 control being tested
Applicability	Most of these controls apply directly to the devices in a GSS (A-130) or network, which need to be looked at together to manage overall risk.
Inherited By	Most applications (a-130) will be supported by specific devices on one or more networks, and inherit risk from those devices.
Diagnostic Responsibility	Who or what will perform the assessment
Defect Metrics	Which defect type or metric will identify related defects
Mitigation Responsibility	Who (person or group) is responsible to deal with defects (including risk acceptance decisions)
Selected?	Will your D/A select and use this test?
Risk Acceptance	What risk is accepted if the control is not selected or does not work perfectly.
Frequency	How frequently the test should be attempted.
Impact	How much impact you D/A assesses a failure of this control creates.

Example Assessment Narrative

Test narrative is to be provided for each row in the CAT tables. Here is an example for CM-02:

- Control CM-02: The organization develops, documents, and maintains under configuration control, a current baseline configuration of the [Scope Limitation for HWAM – HARDWARE devices and optional device hardware sub-components in the] information system.
 - Applicability: All Networks Using ISCM/CDM.
 - Inheritance: Control is inherited by all applications having that device in its accreditation boundary.
 - Determination Statement: The Desired State Inventory is Maintained.
 - Assessment and Diagnosis Responsibility: Automated by CDM
 - Assessment methods:
 - Test 1: Defect Type F1 will through a defect if the Actual and Desired States are not equal. This would occur if the Desired State does not have new devices added.
 - Test 2: Defect Type L7 will throw a defect if authorized inventory is not reviewed, as needed to remove authorizations periodically and/or on an event driven basis.
 - Test 3: Defect Type L3 will throw a defect if a required device is not added to the network and kept there.
 - Related Tests: Add Subcomponents (Defect Type L6)
 - Required Maturity: Level 2 – Capability
 - Mitigation Responsibility: Network Operations: Desired State Manager or Designee.
 - Mitigation Methods: Stated in Defect Type Table.
- NIST review and subsequent adjustment in process

HWAM Moderate Baseline (Part 1)

Control Allocation Table - CAT

Control Item	Applicability	Inherited By	Diagnostic Responsibility	Defect Metrics	Mitigation Responsibility	Selected	Risk Acceptance	Frequency	Impact
CM-02 (1a)	Network	Applications	CDM Check	L7	NetOps and DSM			< 4 days	
CM-02 (1b)	Network	Applications	CDM Check	L7	NetOps and DSM			< 4 days	
CM-02 (1c)	Network	Applications	CDM Check	F1	NetOps and DSM			< 4 days	
CM-02 (7a)	Network	Applications	CDM Check	L1 & L6	NetOps			< 4 days	
CM-02 (7b)	Network	Applications	CDM Check	L1 & L6	NetOps			< 4 days	
CM-03a	Network	Applications	Network ISSO & Application ISSO	Defect Type Table shows which defect types are selected.	Network ISSO & Application ISSO			Event Driven	
CM-03b	Network	Applications	CDM Check	F1 & L2	DSM			< 4 days	

HWAM Moderate Baseline (Part 2)

Control Allocation Table - CAT

Control Item	Applicability	Inherited by	Diagnostic Responsibility	Defect Metrics	Mitigation Responsibility	Selected	Risk Acceptance	Frequency	Impact
CM-03c	Network	Applications	CDM Check	L2	DSM			< 4 days	
CM-03d	Network	Applications	CDM Check	L3	NetOps			< 4 days	
CM-03e	Network	Applications	DSM	Desired State data show data for required period	HWAM Operator			Annually or Event Driven	
CM-03f	Network	Applications	CDM Check	F5	DSM and HWAM Operator			< 4 days	
CM-03g	Network	Applications	CDM Check	L2	DSM			< 4 days	
CM-03 (2)	Network	Applications	CDM Check	F1	DSM			< 4 days	
CM-08 (1)	Network	n/a	CDM Check	F1	NetOps & DSM			< 4 days	
CM-08 (3a)	Network	Applications	CDM Check	F1	NetOps & DSM			< 4 days	

HWAM High Baseline

Control Allocation Table - CAT

Control Item	Applicability	Inherited By	Diagnostic Responsibility	Defect Metrics	Mitigation Responsibility	Selected	Risk Acceptance	Frequency	Impact
AC-19 (5)	Network	Applications	CDM Check	L4	NetOps			< 4 days	
CM-02 (2)	Network	Applications	CDM Check	F1, L2, & L8	DSM			< 4 days	
CM-03 (1a)	Network	Applications	CDM Check	F1, L2, & L8	DSM			< 4 days	
CM-03 (1b)	Network	Applications	CDM Check	F1 & L2	DSM			< 4 days	
CM-03 (1c)	Network	Applications	CDM Check	L9 (assumes L2]	DSM			< 4 days	
CM-03 (1d)	Network	Applications	CDM Check	F1 & L2	DSM			< 4 days	
CM-03 (1e)	Network	Applications	CDM Check	F1 & L2	NetOps & DSM			< 4 days	
CM-03 (1f)	Network	Applications	CDM Report	Verify report is available	HWAM Operator			Annual or Event Driven	
CM-08 (2)	Network	Applications	CDM Check	F1 & L6	HWAM Operator			< 4 days	
CM-08 (4)	Network	Applications	CDM Check	F2	DSM			< 4 days	
SA-12	Network	Applications	CDM Check	L5	NetOps			< 4 days	

Use Case

Overall Process

- Control Selection
 - Impact
 - Selection
 - Risk Acceptance
- Inherit/Customize Test Plan
 - What to Test
 - Assessment/Diagnostic Responsibility
 - Assessment/Diagnostic Frequency
 - Mitigation Responsibility
- Assessment (largely automated)
- Action Tracking (largely automated)
 - Overall Improvement
 - Triggers
 - Individual Actions

Control Selection

- **Impact:** Assess the impact of a failure of each control. This may be an ongoing process based on experience as to whether the control is needed to avoid actual incidents/risks.
- **Selection:** Select which controls to implement, based on the estimated impact of failure, the impact level of the system, key stakeholder risk tolerance, etc.
- **Risk Acceptance:**
 - For selected controls: Document the estimated result of controls not currently working, and whether this can be accepted.
 - For non-selected controls: Document any factors other than system impact level and control impact that affected the decision not to select the control.
 - This provides adequate documentation of Control Selection.

Inherit/Customize Test Plan

- **What to Test**
 - Go through the defect type tables and select which defect types to implement to assess the controls selected.
 - A valid choice may be not to implement non-core defect checks listed for a control.
 - Justify which of the non-core defect tested will not be selected, and why.
- **Assessment/Diagnostic Responsibility**
 - We expect minimal need to customize these columns, but especially in non-federal networks, this may vary more.
- **Assessment/Diagnostic Frequency**
 - For Federal Civilian networks this is specified by the CDM program. Others may want to customize this.
- **Mitigation Responsibility**
 - Categories provided are designed to be more specific than the NIST roles, indicating who the NIST role would delegate to.
 - These can be easily customized by the organization.
- The assessment narratives may then be adjusted accordingly
Completing these items creates an adequate assessment plan.

Assessment (Largely Automated)

- At least for HWAM, the assessment can be largely automated.
- This includes all rows in the CAT tables where the frequency is “< 4 days.”
- Note: The dashboard can display risk and mitigation to-do lists by
 - Supported application
 - For the whole network
 - And by other useful risk management categories

Action Tracking (Largely Automated)

- Overall Improvement: The dashboard will show total risk and average risk per object, which a timeline showing overall improvement.
- Triggers: The DAA can establish control limits to be tracked by the system to trigger danger alerts. For example:
 - Risk per object exceeds some value (control limit)
 - Risk per object goes up n days in a row (suggesting a trend)
 - Risk per object jumps up more than x in one day (suggesting a sudden change).
- Individual Actions
 - Those taking action are provided a to-do list of mitigations for defect for which THEY are responsible, prioritized with the worst problems first.
 - When those are addressed, they disappear in the next round of testing, without additional manual reporting.
- POA&M entries can be limited (mostly) to dealing with triggers.

END OF SECTION

Continuous Diagnostics and Mitigation (CDM)

Module 1 – CDM Overview
Part 8 – Human Factors

Topics

1. **Why Human Factors?**
 - Motivating Action
 - The Right Action
 - Theory X or Theory Y
2. **Stakeholder Identification**
3. **Factors to Manage**
 - Collection System Quality (Sensors and Tools)
 - D/A Scoring/Grading
 - D/A Defect Types
 - Risk Transfers
 - Helping the Weak (Tiger Teams)
4. **“Pilot Phase”**
5. **Generating Friendly Competition**
6. **Ensuring Objectivity, Fairness, and Transparency**
7. **D/A Help Desk Participation**
8. **Monthly Cyber-Scope Reporting**

Why Human Factors?

- It often seems that CDM is focused on technology
 - Tools/Sensors
 - Dashboard
 - Operations and Maintenance
- All of that is a necessary part of CDM.
- But the true focus of CDM is to use that data to help PEOPLE make good decisions and take action to improve security, from System Admins to Agency Heads.
- **The technology only works when it is used to properly motivate and enable people to take the best actions to protect information.**

Management Theory Y

- Douglas McGregor of MIT initiated the idea that there are (broadly) two management approaches, Theory X and Theory Y.
- Paraphrasing to McGregor's observations¹:
 - Theory Y managers assume people want to perform well and just need help.
 - The CDM system is designed to provide such help. Today, it's widely accepted that good management requires a Theory Y orientation.
 - Theory X managers, by contrast, tend to assume that people are lazy and unreliable.
 - Consequently, these managers feel the need to put in all kinds of control systems and time clocks — all the paraphernalia of bureaucracy at its worst.
- **Managing with Theory X will generally prevent CDM from working, because line security workers will feel betrayed by management.**

1: <http://mitsloan.mit.edu/faculty/spotlight/pioneered.php>

Stakeholder Identification

- Each implementing organization must identify the particular functions that require CDM data to perform, and ensure that they get that data and know what to do with it.
- This might include:
 - Getting Dashboard Access
 - Simple guidance on using the off-the-shelf Dashboard
 - Developing Custom Reports
- The CIO and CISO will want to allocate some effort to performing this work.

Factors to Manage: Data Quality

- Some data quality issues may be the responsibility of the CMaaS provider.
 - Automating the right collectors to cover all devices and defect checks with sufficient speed to enable effective mitigation
 - Departments and agencies will need to allocate some effort to oversight.
 - The CDM Office will support this oversight role.
- Other data quality issues are a D/A responsibility. For example,
 - Sensors need credentials to do authenticated scans.
 - Desired state data must be current.
 - Sensors must be able to get through firewalls.
 - The D/A will need to allocate some effort to keep essential data available.
- In some cases CDM sensors may not perform as expected. The CDM office will help address any unexpected data quality problems with the CMaaS provider and D/As on a case-by-case basis.
- **Critical Success Factor:** Too many false positives will poison the well.

Factors to Manage: Scores/Grades

- Scores and grades should be used to motivate and assess performance.
- Scores and grades have to be viewed as fair, transparent and objective.
- There needs to be an established process to validate fairness, transparency, and objectivity, and for concerns to be raised.
- There also needs to be a process for the resolution of such concerns.
 - One method is implementing a webpage to collect issues, with a team assigned to investigate and find solutions.
 - Another method is coordinate meeting to report changes and get feedback.
 - List-serves and blogs for stakeholders can be used to watch for issues, find solutions, and report those solutions.
 - It may also be effective to develop an FAQ with relevant issues and solutions.

Be like Mr. Spock!
Depersonalize the issue AND the response.
Be willing to fix problems.

Factors to Manage: Defect Checks

- Define defect checks realistically: Do *not* create a policy (defect checks) if the policy cannot be effectively and efficiently implemented.
- As defect checks are defined, it is essential to make sure they can be addressed. For example:
 - Why hold people responsible for a CVE that has no mitigation?
 - Why demand a repair that will break a mission critical application?
- Work with operators to identify problems and find actual solutions, or consider eliminating the defect check and accepting the risk.

Factors to Manage: Risk Transfers

- In operational environments, responsibility is often shared for different aspects of a device.
- In implementing capabilities, it is important to find short-cuts to decide who is responsible for a device.
- However, it is important to recognize that the device manager (say, a LAN Admin) may not be responsible for all the risk on the device:
 - Classic example: We need to patch JRE, but a department wide application will break if we do. What do we do?
 - CDM does not provide exceptions, because the risk is still there.
 - But there is a solution: The application needs to work with a safer version of JRE.
 - So, the risk for unpatched JRE problems that would break the application should be transferred to the application technical manager/application owner.
 - This is then tracked as a POA&M until fixed.
 - Then risk can be transferred back to the LAN Admin.
- The dashboard supports this process. **The process is vital to a sense of fairness.**

Factors to Manage: Risk Transfers

(Cont.)

The CIO or CISO will need to devote some resources to facilitate this process:

- Receive requests
- Adjudicate
- Issue transfers
- Track progress

Factors to Manage: When All Else Fails!

- Inevitably, a few security workers/teams will not have acceptable performance.
- What can be done?
 - Provide a mentor (aka tiger team) to help them get going.
 - Provide central mentors and/or ask them to get help from a successful colleague.
 - Ask their managers to make sure they know to give security more priority (this may help managers too).
 - Ask their managers to make sure they have the staff, training, resources necessary to succeed.
 - Finally, consider reassigning responsibility.

Factors to Manage: When All Else Fails!

(Cont.)

- In one large agency, some lower performers argued they did not have as many staff as others, so they “couldn’t” succeed.
- If this were true, management would need to reallocate staff.
- The CISO did a sample study to compare staffing in high and low performers with large and small pools of devices.
 - The study found no difference in performance based on staff/device, level of training, CISO assessment of staff ability, or any other HR factor.
 - This was communicated, and security workers asked if they could suggest another study to show the correlation. They did not.
 - Anecdotal visits to high and low performing teams suggested that low performance was clearly linked to low dashboard use.
 - Mentoring on dashboard use to low performers showed that 80% significantly reduced risk within 6 weeks.
- The CDM Program Office will work with D/As to find simple solutions.

Shake-Down/Pilot Phase

- Inevitably, there will be false positives during the CDM roll-out.
- If grading begins immediately, people will feel abused by the false positives, and may resist implementation.
- Instead, a pilot phase can be adopted to:
 - Only send grades and scores to those directly responsible to fix things.
 - Ask them to report false positives and fairness issues.
 - Commit to fix problems and complete that commitment.
 - Begin reporting to management when participants agree the data is clean and fair.
- Surprisingly, people may be more motivated to improve before grades are reported to management. That gives them time to look good when grades are first reported.

Friendly Competition

- What motivates people to improve?
 - For some, it's friendly competition to stay with the group.
 - Others want to see their performance improve (competing with themselves).
 - Top performers improve to stay ahead of others.
 - Some may improve because the CDM system enables them to do it.
- It is possible to link contractor compensation to adequate performance and improvement
- Use friendly competition to improve performance.
- Use Theory Y (Not X).

Helpdesk Participation

- CDM will cause helpdesk tickets,
 - Particularly at first
 - Particularly about data quality and fairness.
- The CMaaS provider can provide a script for the local help desk and guidance on when/where to escalate:
 - Level 1 – Local D/A Helpdesk(s)
 - Level 2 – CMaaS Helpdesk
 - Level 3a – CDM program for Policy/Dashboard Issues
 - Level 3b – Tool providers for Sensor/Tool Issues
- **Be prepared for helpdesk tickets.**

Monthly CyberScope Reporting

- The first version of the dashboard is expected to support monthly CyberScope reporting.
- There will be some work to find standard identifiers for object-types to report.
- CDM program office and the CMaaS contractor will help with this.
- This should reduce workload in this area.

Q&A

Any future questions can be sent to:

cdm.fnr@hq.dhs.gov